SCTR's ,PUNE INSTITUTE OF COMPUTER TECHNOLOGY,PUNE-411043
DEPARTMENT OF COMPUTER ENGG.
ACADEMIC YEAR: 2020-21(SEM 1)

SDL MINI PROJECT REPORT

Group Members:

31161 – Shreya
31164 – Prathamesh S.
31168 – Tanushree
31170 – Prathamesh T.

TITLE

ColabCode

PROBLEM STATEMENT

Design an Android app to simplify the process of Final Year project allocation. Students and teachers can easily join groups according to their domain and subdomain of interest.

MOTIVATION

1. BE project allocation is a tedious work
2. App simplifies the whole process of allocation of Students and Teachers according to their domain of interest by listing all users at one place.
3. Reduces time and manual work

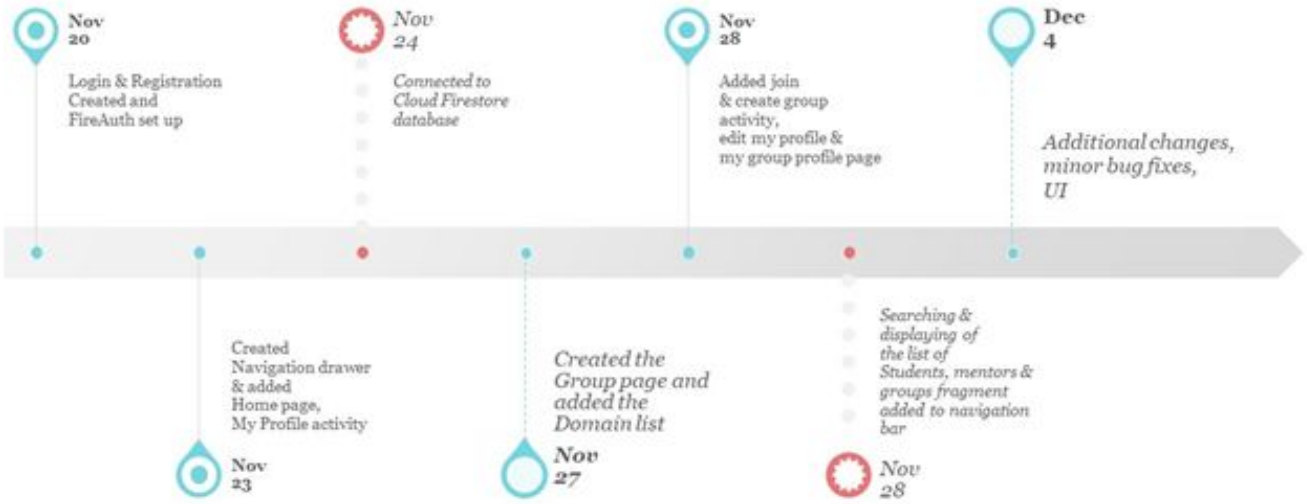
ALGORITHMS

https://github.com/TanushriPatil/SDL-Mini-Project

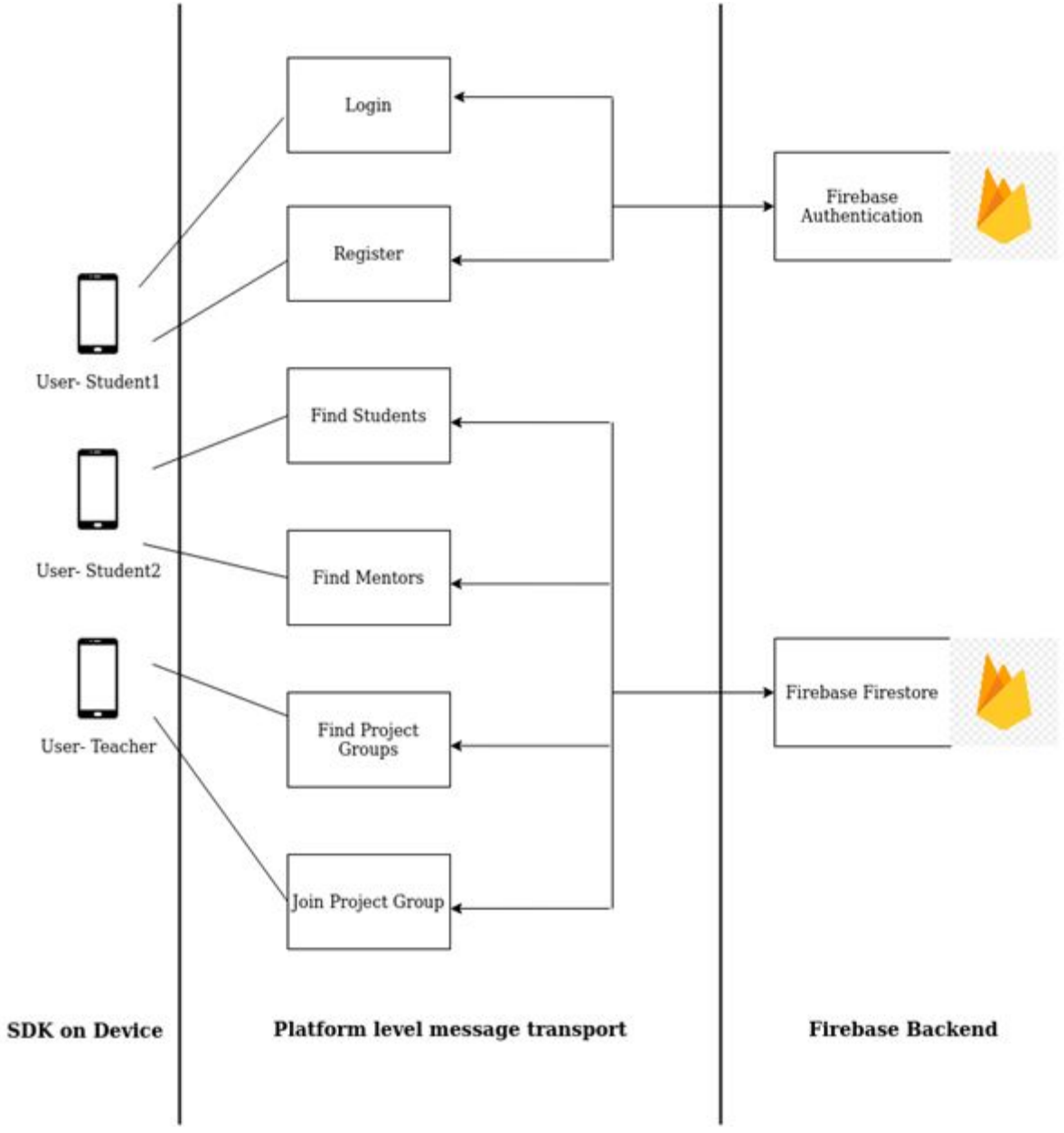
REQUIREMENTS

1. Hardware:
   PC, i7 processor, 8GB RAM
2. OS:
   Windows/ Ubuntu
3. Software:
   Android Studio 4.1.1
4. Database:
   Cloud Firestore

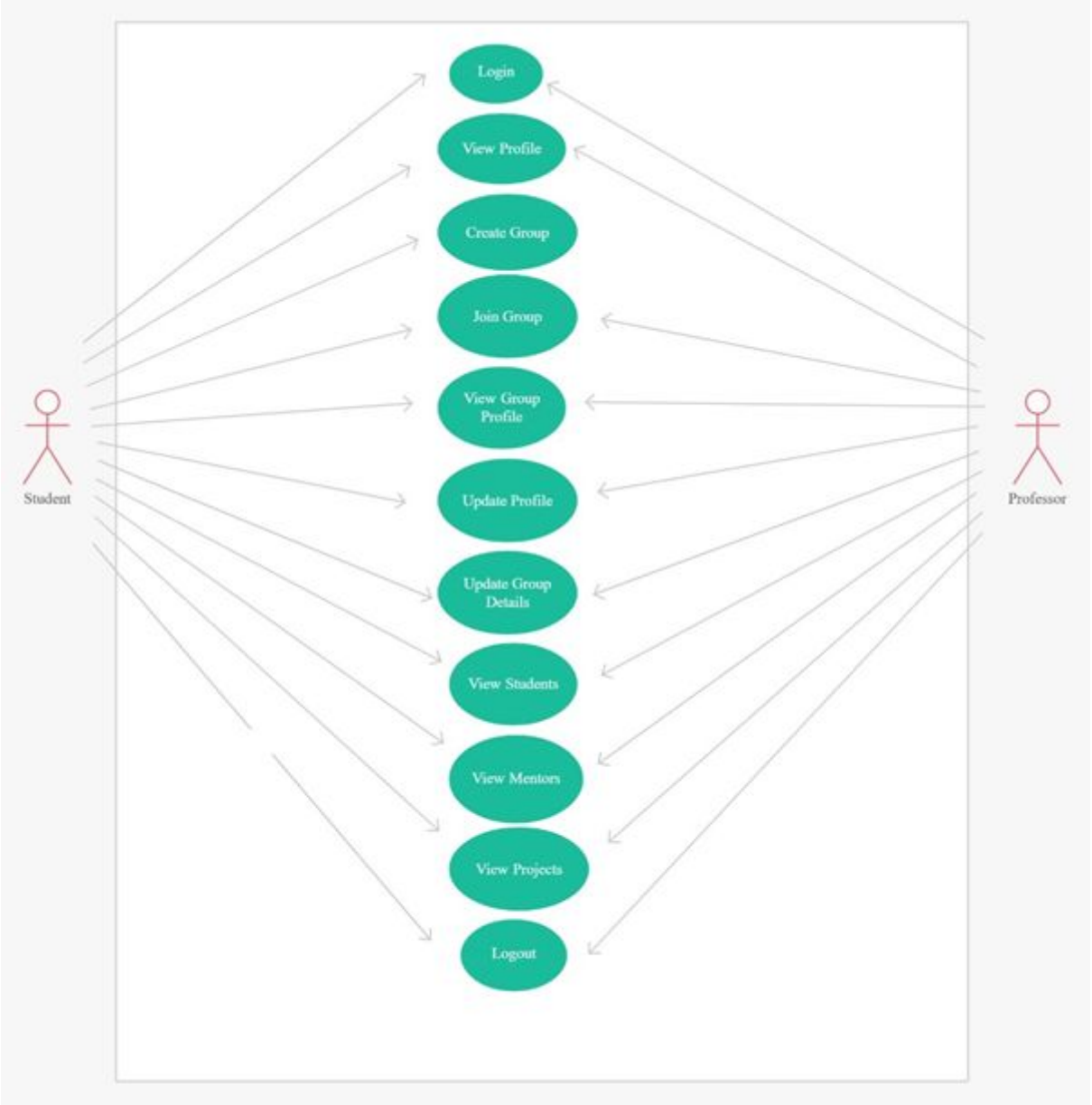
PROJECT TIMELINE



ARCHITECTURE DIAGRAM



HIGH LEVEL DESIGN (USE CASE)



FUTURE SCOPE

- Integration with Chat Bot
- Group Members can chat with each other and Mentor
- Invite Request can be sent to Student/ Mentor/ Group
- Submission of Projects through this app
- Recommendation System to suggest Group Members and Mentor using ML Algorithm

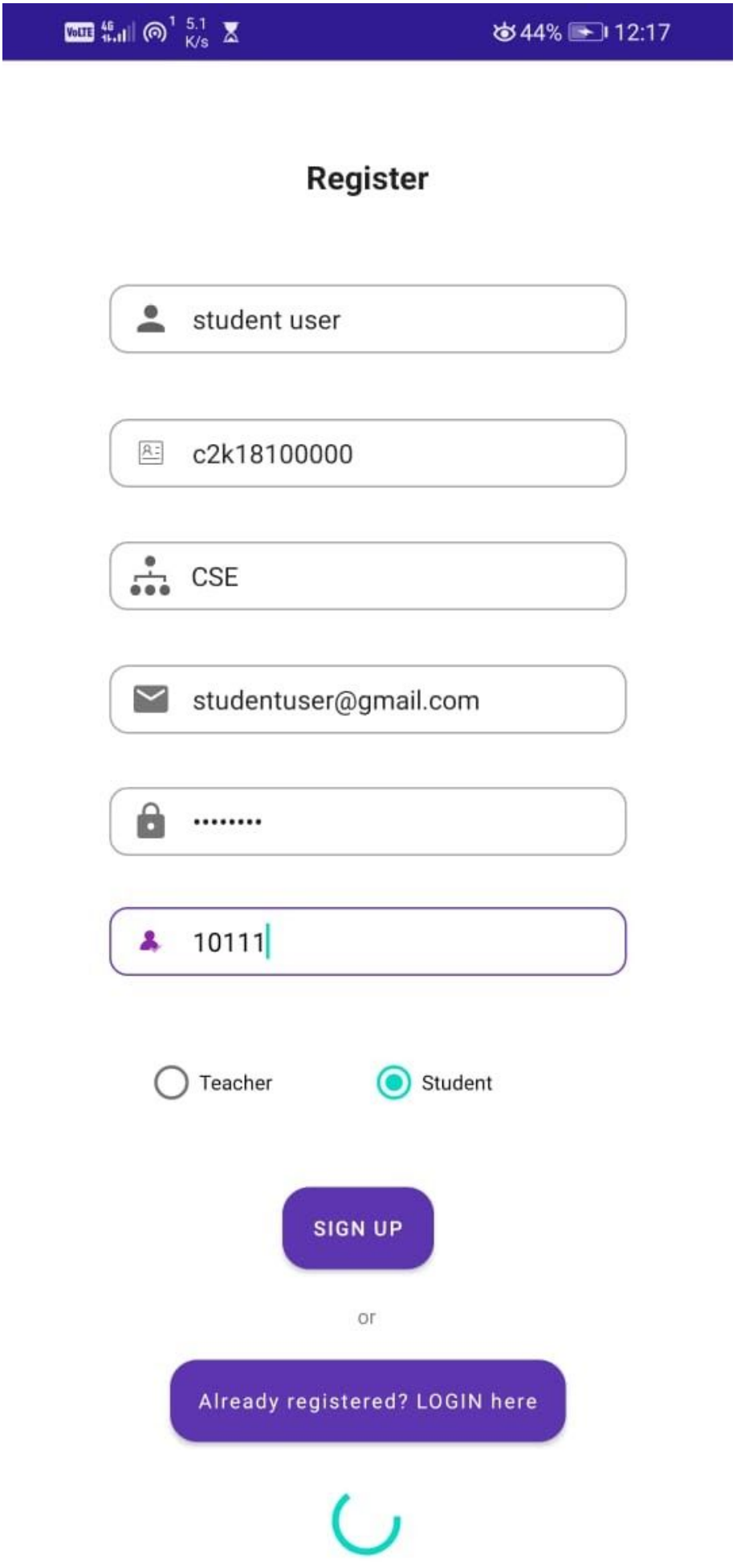
CONCLUSION

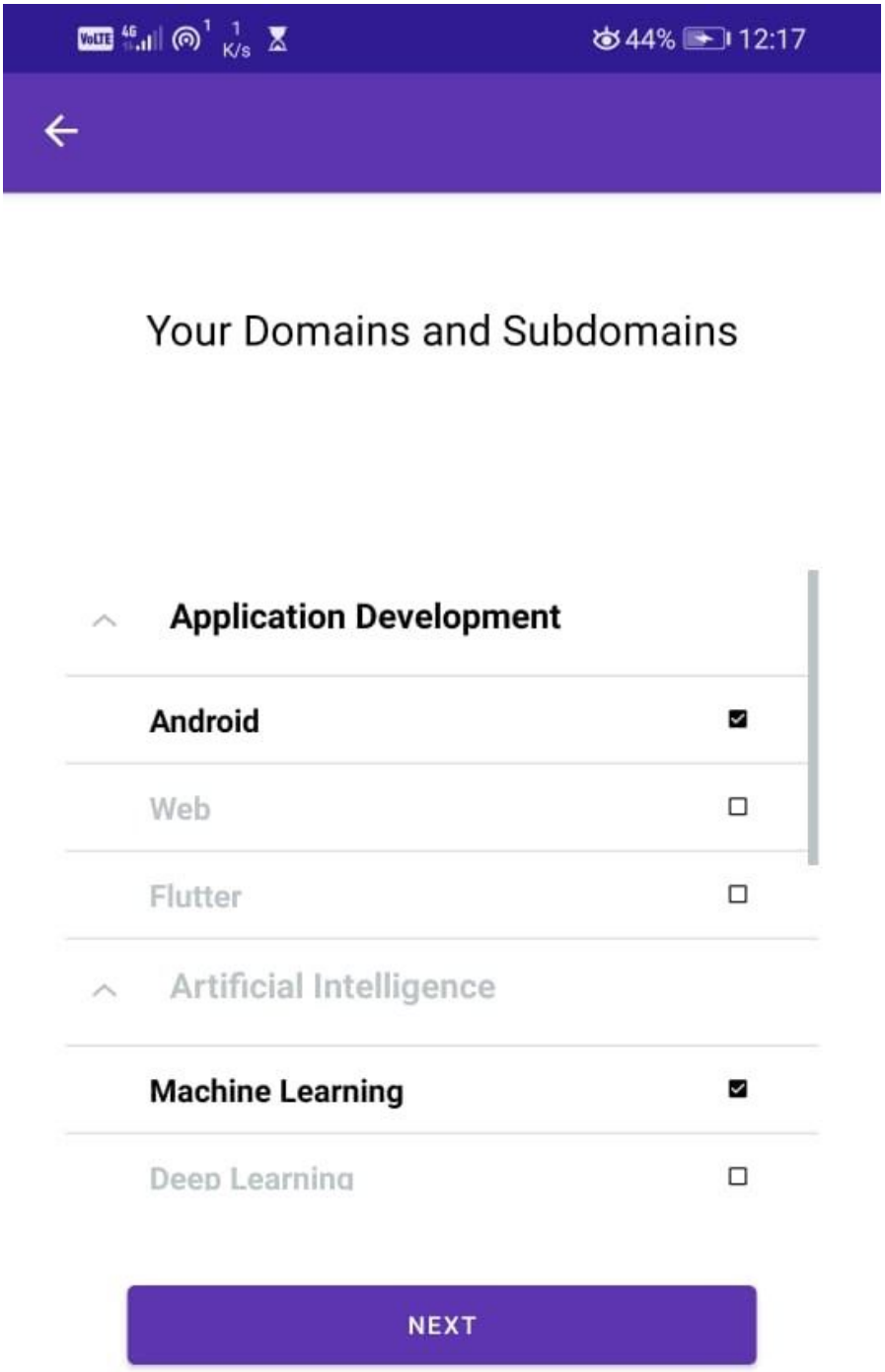
We successfully created an app that will make it easier for the students and teachers to view the domains, and groups and join them easily.

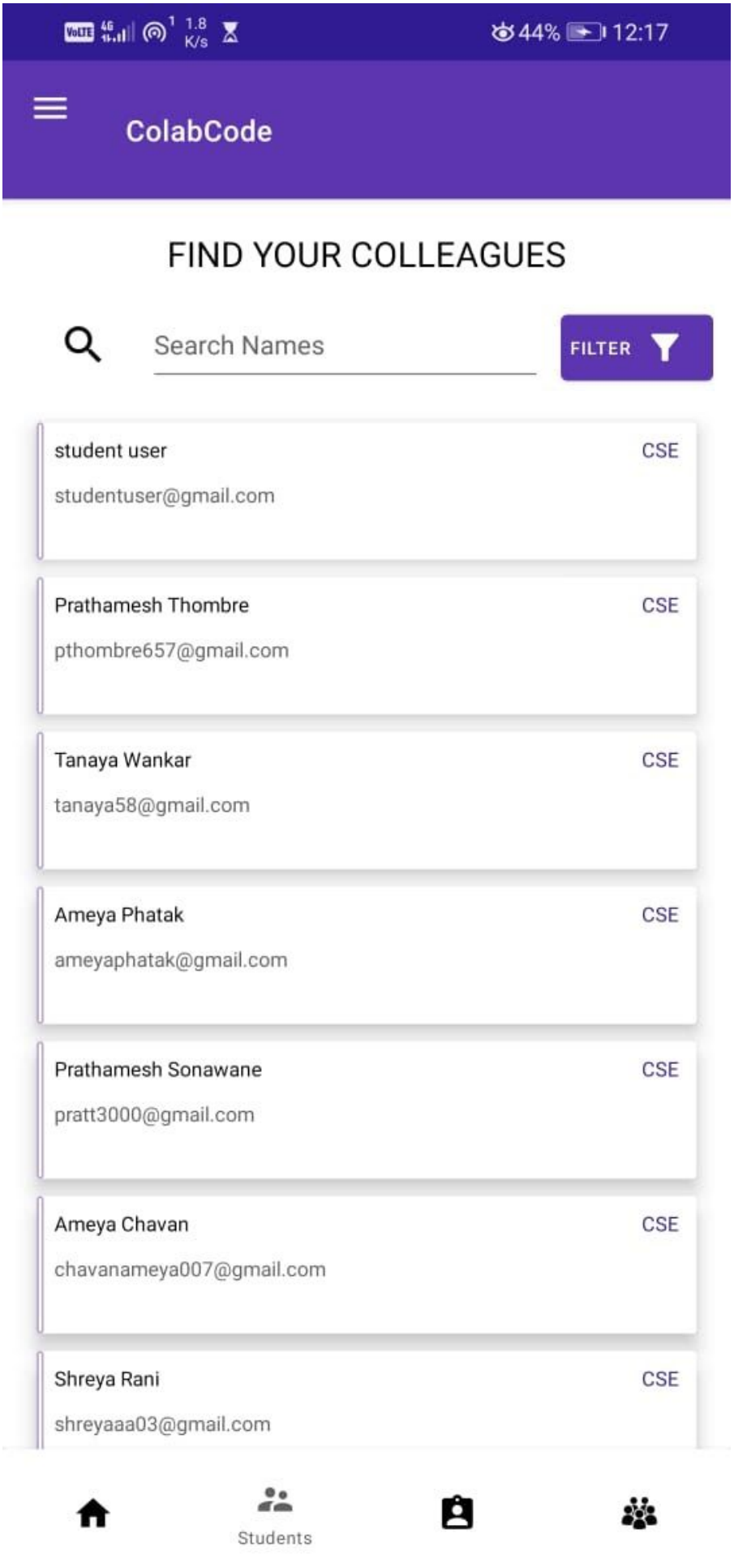
REFERENCES

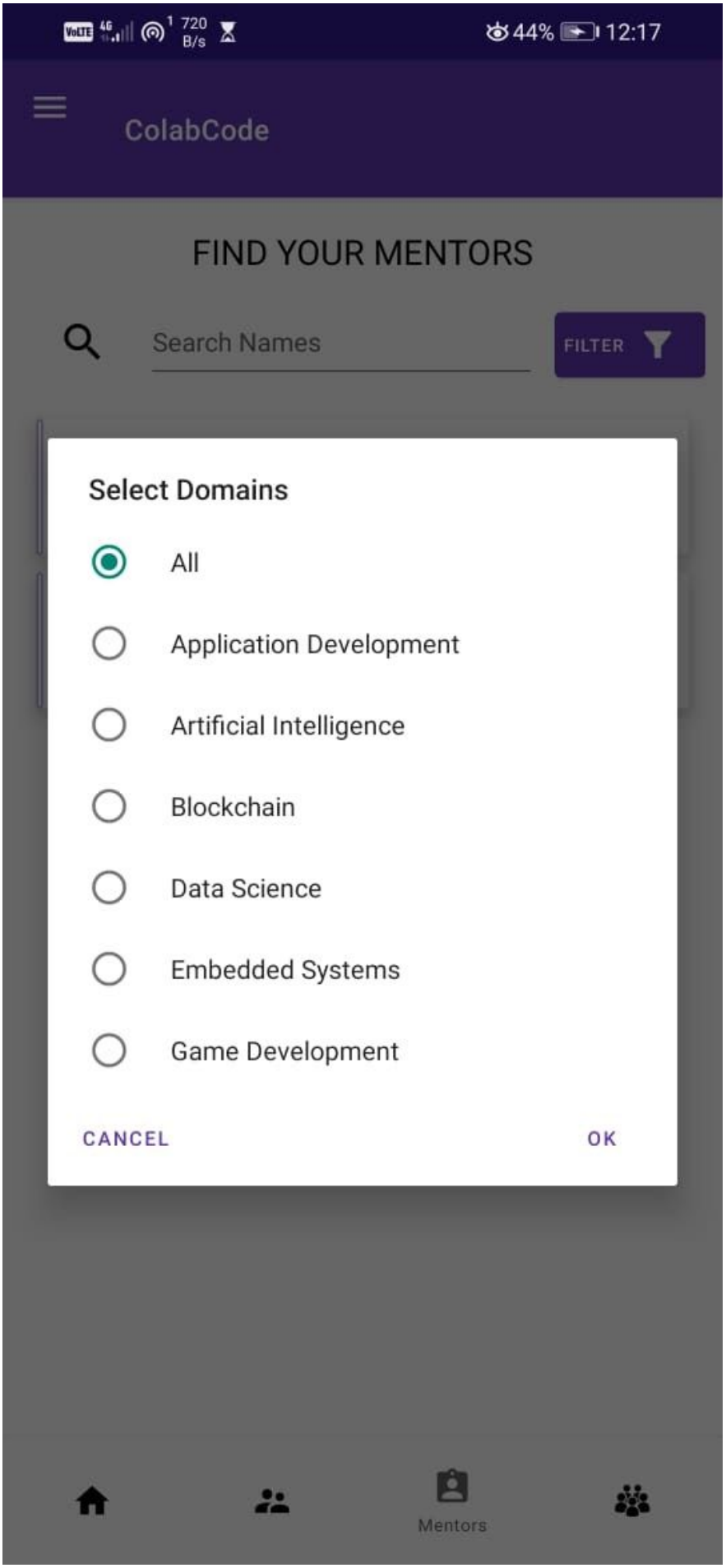
1. https://developer.android.com/studio/intro
2. https://firebase.google.com/docs

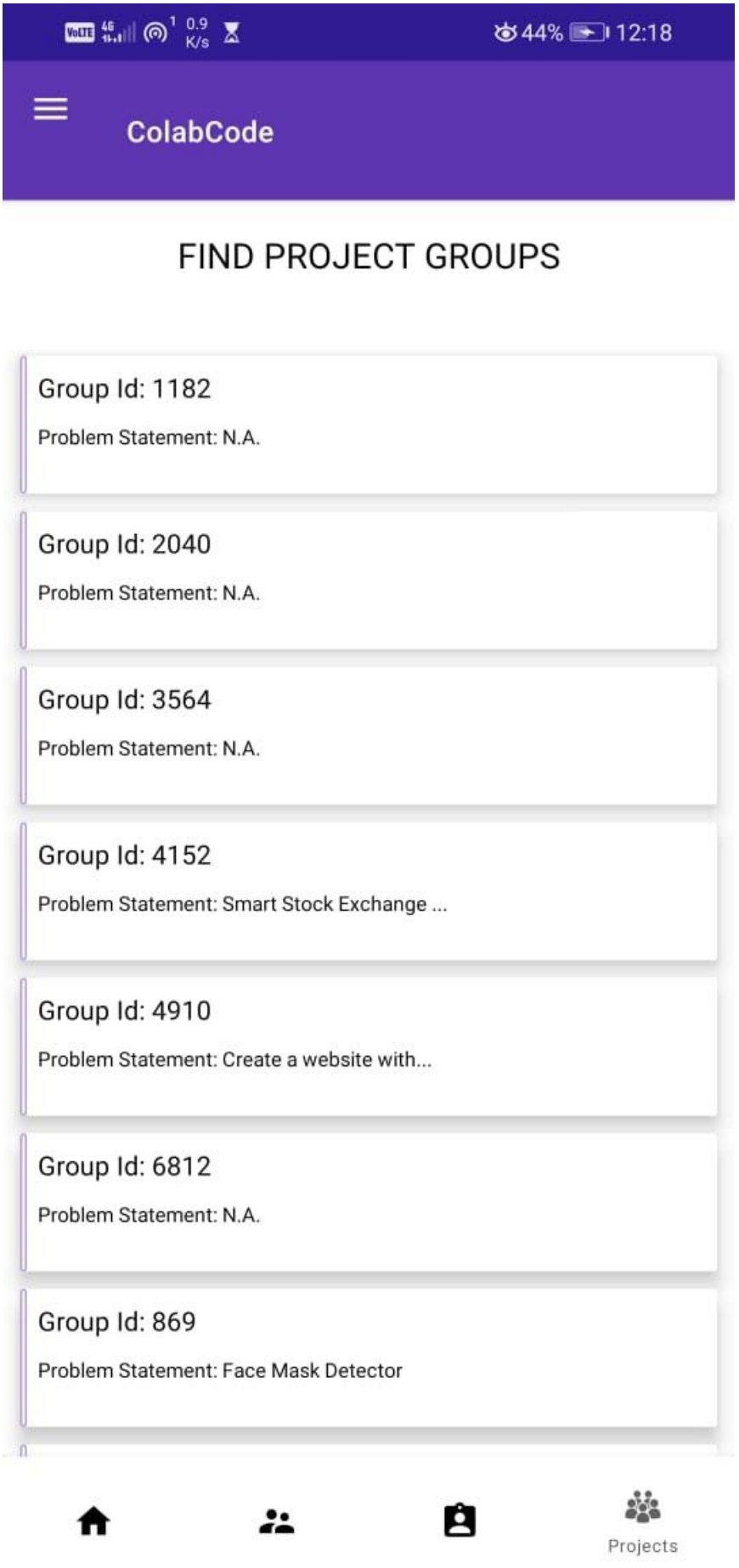
UI's:

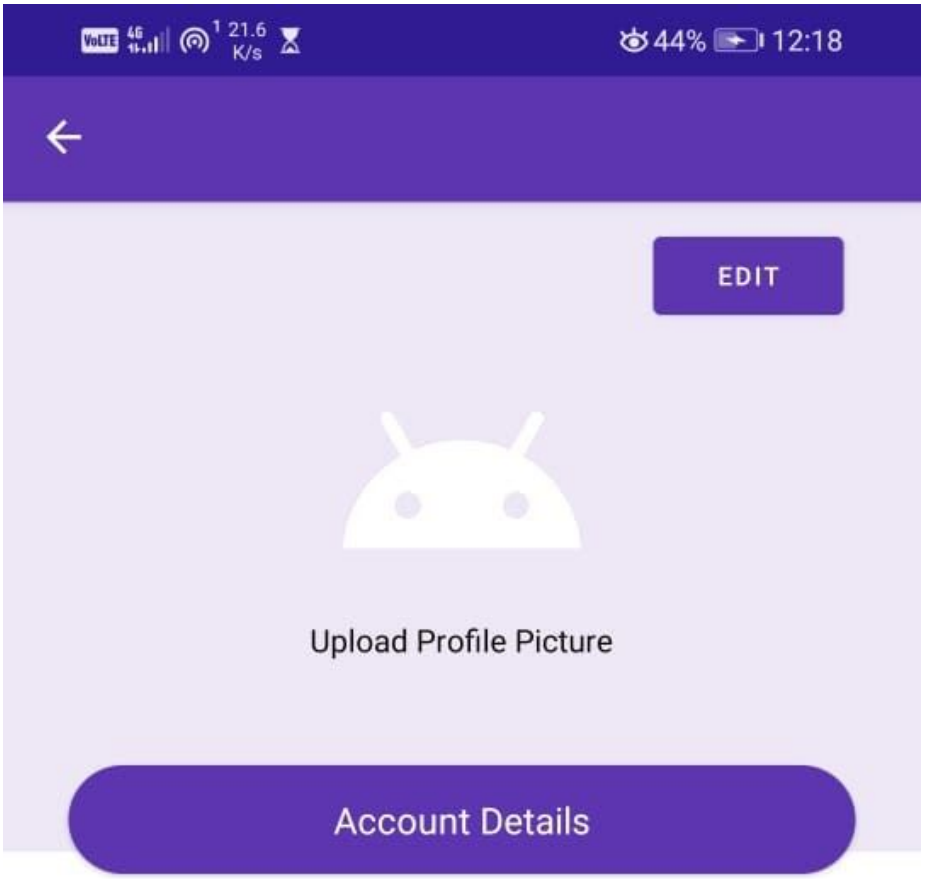

44% 12:17
Register
student user
c2k18100000
CSE
studentuser@gmail.com
10111
Teacher
Student
SIGN UP
or
Already registered? LOGIN here


44% 12:17
Your Domains and Subdomains
Application Development
Android
Web
Flutter
Artificial Intelligence
Machine Learning
Deep Learning
NEXT


44% 12:17
ColabCode
FIND YOUR COLLEAGUES
Search Names
FILTER
student user
CSE
studentuser@gmail.com
Prathamesh Thombre
CSE
pthombre657@gmail.com
Tanaya Wankar
CSE
tanaya58@gmail.com
Ameya Phatak
CSE
ameyaphatak@gmail.com
Prathamesh Sonawane
CSE
pratt3000@gmail.com
Ameya Chavan
CSE
chavanameya007@gmail.com
Shreya Rani
CSE
shreyaaa03@gmail.com
Students


44% 12:17
ColabCode
FIND YOUR MENTORS
Search Names
FILTER
Select Domains
All
Application Development
Artificial Intelligence
Blockchain
Data Science
Embedded Systems
Game Development
CANCEL
OK
Mentors

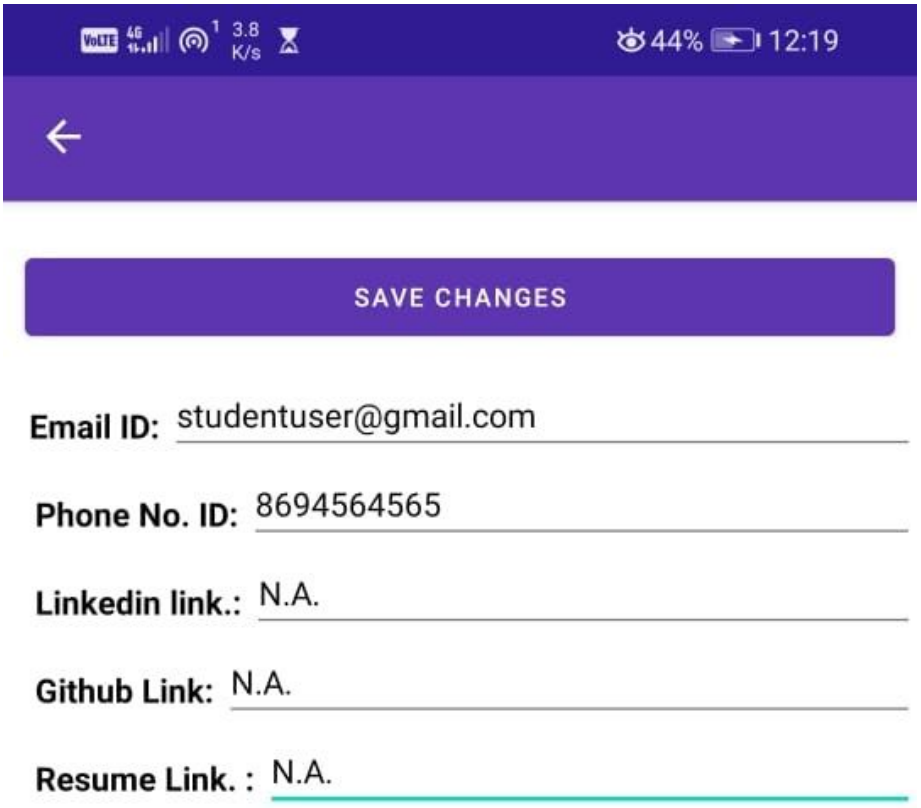

0.9 K/s
44% 12:18
ColabCode
FIND PROJECT GROUPS
Group Id: 1182
Problem Statement: N.A.
Group Id: 2040
Problem Statement: N.A.
Group Id: 3564
Problem Statement: N.A.
Group Id: 4152
Problem Statement: Smart Stock Exchange ...
Group Id: 4910
Problem Statement: Create a website with...
Group Id: 6812
Problem Statement: N.A.
Group Id: 869
Problem Statement: Face Mask Detector
Projects

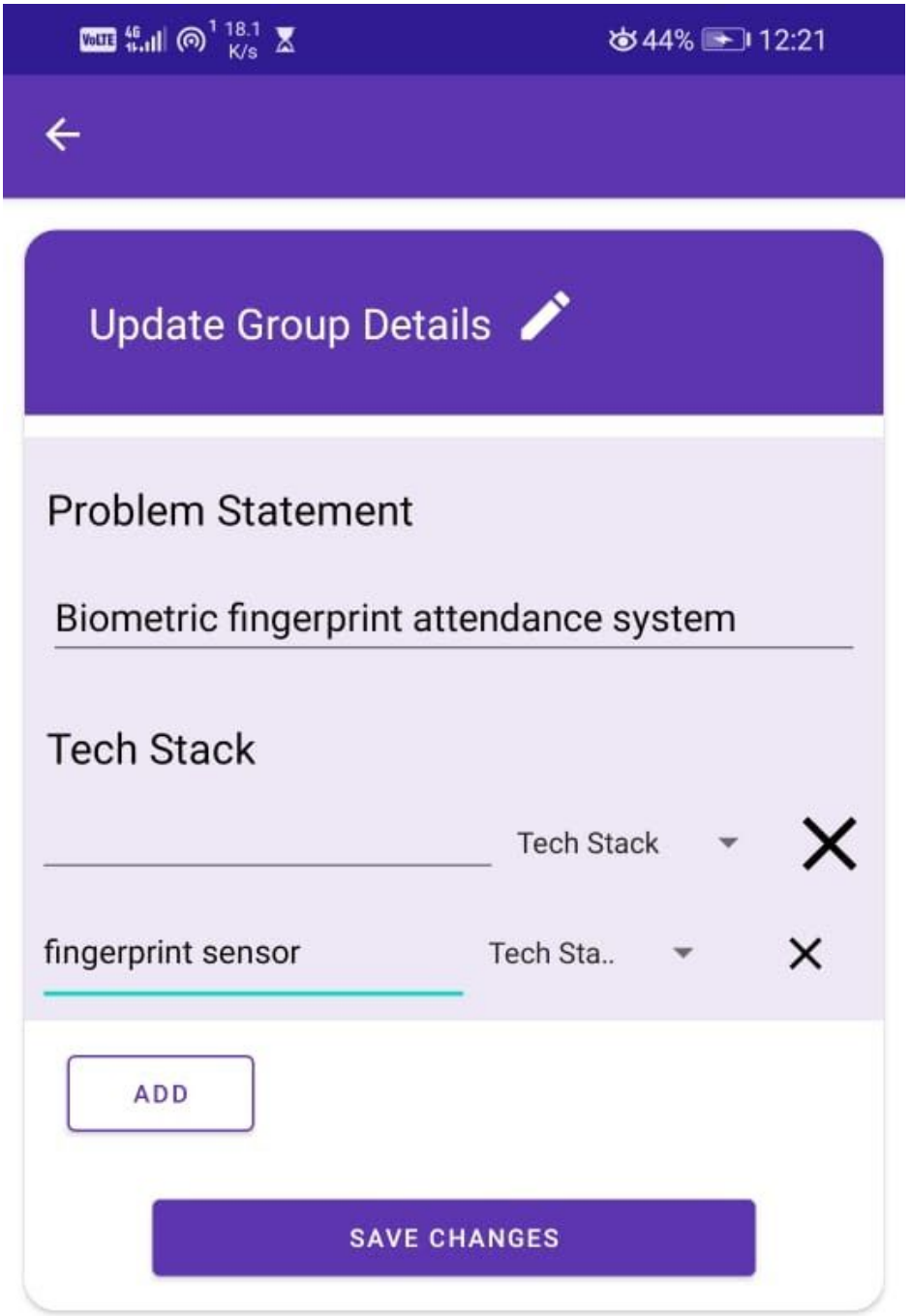


Application Development : Android

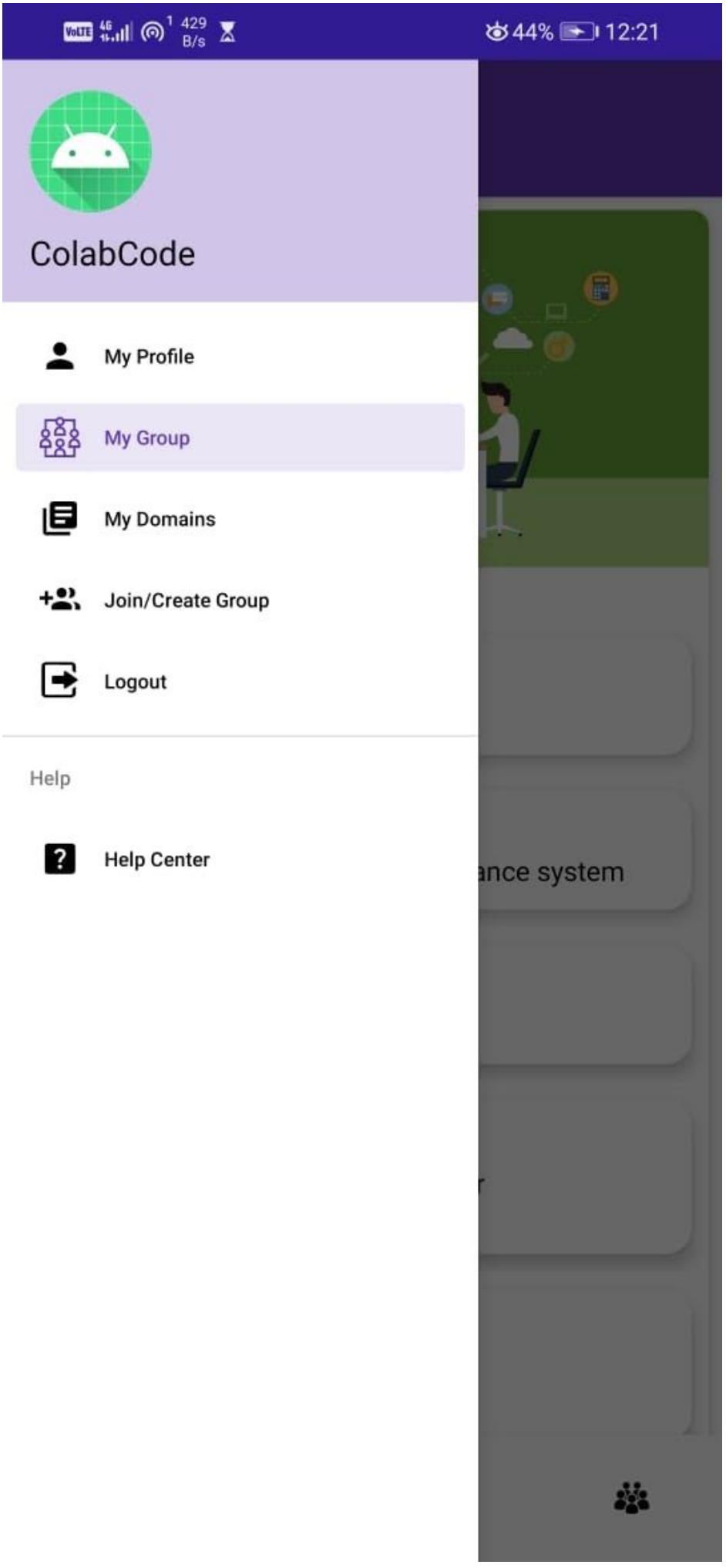
Artificial Intelligence : Machine Learning

Name	student user
Reg ID	C2K18100000
Email ID	studentuser@gmail.com
Phone no.	N.A.
Branch	CSE
Roll no.	10111
Role	Student
Group id	6812
Linkedin link	N.A.
Github link	N.A.
Resume link	N.A.


3.8
K/s
44%
12:19
SAVE CHANGES
Email ID: studentuser@gmail.com
Phone No. ID: 8694564565
Linkedin link.: N.A.
Github Link: N.A.
Resume Link. : N.A.


18.1 K/s
44% 12:21
Update Group Details
Problem Statement
Biometric fingerprint attendance system
Tech Stack
Tech Stack
fingerprint sensor
Tech Sta..
ADD
SAVE CHANGES

44%
12:21
ColabCode
Group ID
6812
Problem Statement
Biometric fingerprint attendance system
Mentor ID
Members
C2K18100000 - student user
Tech Stack
fingerprint sensor
Home


44% 12:21
ColabCode
My Profile
My Group
My Domains
Join/Create Group
Logout
Help
Help Center
ance system

LOGIN

teacheruser@gmail.com

••••••••••

LOGIN

Or

CREATE A NEW ACCOUNT

45% 12:24

ColabCode

Group ID

6812

Problem Statement

Biometric fingerprint attendance system

Mentor ID

EMP12346555 - teacher user

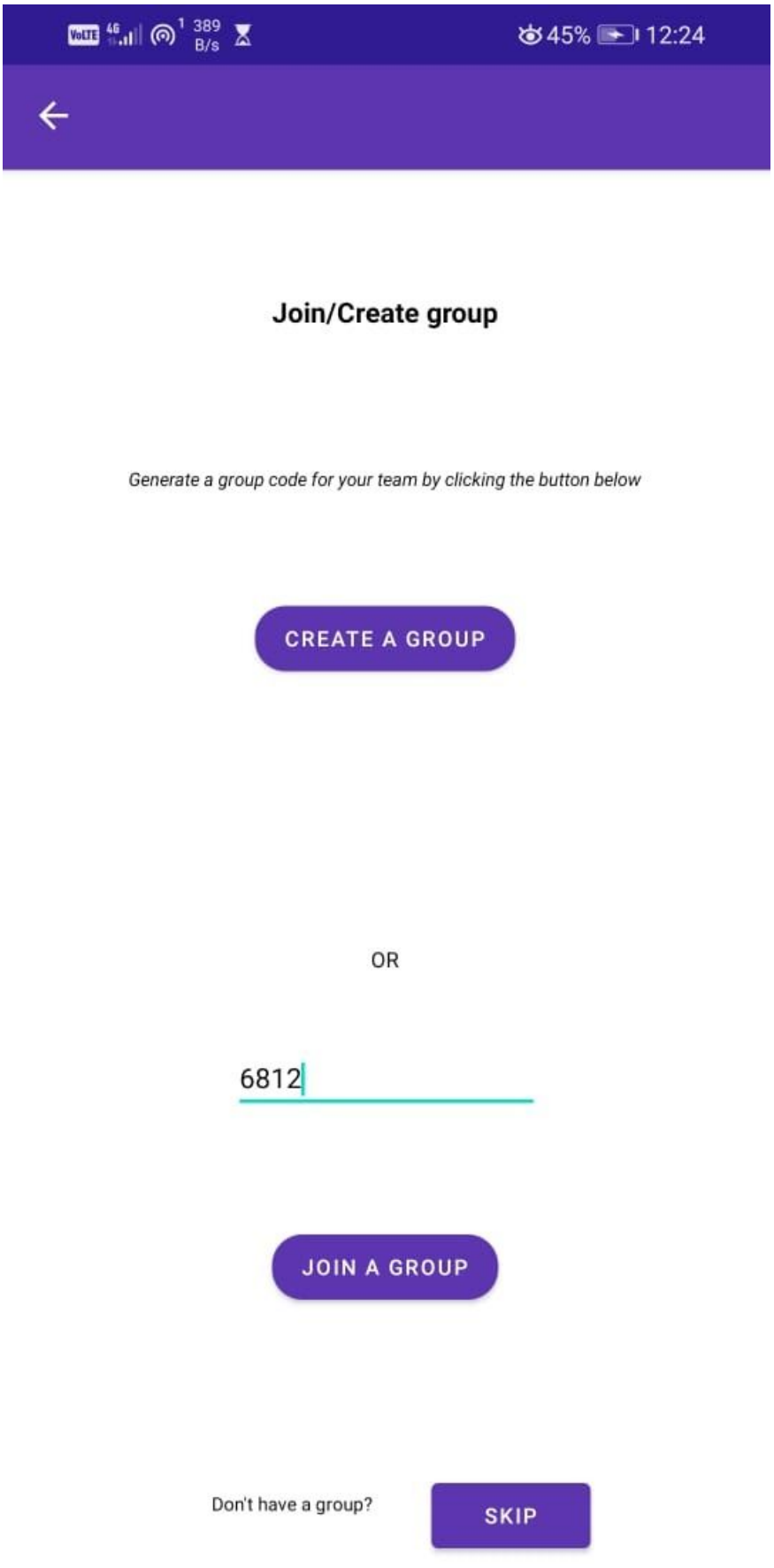
Members

C2K18100000 - student user

Tech Stack

fingerprint sensor

Home


389 B/s
45% 12:24
Join/Create group
Generate a group code for your team by clicking the button below
CREATE A GROUP
OR
6812
JOIN A GROUP
Don't have a group?
SKIP

Join/Create group

Generate a group code for your team by clicking the button below

CREATE A GROUP

Group code: 6812

Share it with group members only!!

OR

Enter Group Code

JOIN A GROUP

You're a member of team: 6812

SOURCE CODE

MainActivity.java

```
package com.example.teproject;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.fragment.app.Fragment;

import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.auth.FirebaseAuth;

public class MainActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {

    private DrawerLayout drawer;
    NavigationView navvw;
    BottomNavigationView bottomNav;
    private static final String TAG = "MainActivity";
    private static final String CALLING_TAG = "MainActivity";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        drawer = findViewById(R.id.drawer_layout);

        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this, drawer, toolbar,
R.string.navigation_drawer_open, R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.setDrawerIndicatorEnabled(true);
        toggle.syncState();
        navvw = findViewById(R.id.nav_view);
        navvw.setNavigationItemSelectedListener(this);
```

```
        bottomNav = findViewById(R.id.bottom_navigation);
        bottomNav.setOnNavigationItemSelectedListener(navListener);

        getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
new HomeFragment()).commit();

    }

    private BottomNavigationView.OnNavigationItemSelectedListener navListener =
            new BottomNavigationView.OnNavigationItemSelectedListener() {
                @Override
                public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                    Fragment selectedFragment = null;
                    switch (item.getItemId()) {
                        case R.id.nav_home:
                            selectedFragment = new HomeFragment();
                            break;
                        case R.id.nav_students:
                            selectedFragment = new ListUsersFragment("Students");
                            break;
                        case R.id.nav_teachers:
                            selectedFragment = new ListUsersFragment("Teachers");
                            break;
                        case R.id.nav_projects:
                            selectedFragment = new ProjectsFragment();
                            break;
                    }

getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
                            selectedFragment).commit();
                    return true;
                }
            };

    @Override
    public void onBackPressed() {
      if (drawer.isDrawerOpen(GravityCompat.START)){
          drawer.closeDrawer(GravityCompat.START);
      } else{
         super.onBackPressed();
      }
    }

    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()){
            case R.id.nav_logout:
```

```
            FirebaseAuth.getInstance().signOut();
            startActivity(new Intent(getApplicationContext(), Login.class));
            finish();
            break;
        case R.id.nav_profile:
            Intent profileIntent = new Intent(getApplicationContext(), myprofile.class);
            profileIntent.putExtra("caller", CALLING_TAG);
            startActivity(profileIntent);
            break;
        case R.id.nav_group_profile:
            Intent groupIntent = new Intent(getApplicationContext(), GroupProfile.class);
            groupIntent.putExtra("activity", TAG);
            startActivity(groupIntent);
            break;
        case R.id.nav_domain:
            Intent intent = new Intent(getApplicationContext(), MyDomain.class);
            intent.putExtra("activity", TAG);
            startActivity(intent);
            break;
        case R.id.nav_joingroup:
            startActivity(new Intent(getApplicationContext(), joingroup.class));
            break;

    }
    return true;
  }

}
```

Login.java

```
package com.example.teproject;

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

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
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 com.google.firebase.auth.FirebaseAuth;

public class Login extends AppCompatActivity {

    EditText mEmail, mPassword;
    Button mLoginbtn, mSignUpbtn;
    ProgressBar progressBar;
    FirebaseAuth fAuth;

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

        mEmail = findViewById(R.id.lemail);
        mPassword = findViewById(R.id.lpassword);
        mLoginbtn = findViewById(R.id.lLoginbtn);
        mSignUpbtn = findViewById(R.id.lSignUpbtn);
        fAuth = FirebaseAuth.getInstance();
        progressBar = findViewById(R.id.progressBar2);

        mSignUpbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getApplicationContext(), Register.class));
            }
        });

        mLoginbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email = mEmail.getText().toString().trim();
                String pass = mPassword.getText().toString().trim();

                if(TextUtils.isEmpty(email)){
                    mEmail.setError("Email is required.");
                    return;
                }
                if(TextUtils.isEmpty(pass)){
                    mPassword.setError("Password is required.");
                    return;
                }
                if(pass.length() < 5){
                    mPassword.setError("Password must be >= 5 characters");
                    return;
                }

                progressBar.setVisibility(View.VISIBLE);
```

```
            //authenticate the user

            fAuth.signInWithEmailAndPassword(email, pass).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if(task.isSuccessful()){
                        Toast.makeText(Login.this, "Successfully Logged In",
Toast.LENGTH_SHORT).show();
                        startActivity(new Intent(getApplicationContext(), MainActivity.class));
                    } else{
                        Toast.makeText(Login.this, "Error..! "+ task.getException().getMessage(),
Toast.LENGTH_SHORT).show();
                    }
                }
            });
        }
    });



    }

    @Override
    public void onBackPressed() {

    }
}
```

Register.java

```
package com.example.teproject;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.view.ViewOutlineProvider;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.EditText;
```

```
import android.widget.ProgressBar;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
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 com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;

public class Register extends AppCompatActivity {

    EditText mName, mEmail, mPassword, mBranch,mRegistrationID, mRollno;
    Button mSignUpBtn, mLoginBtn;
    FirebaseAuth fAuth;
    ProgressBar progressBar;
    RadioGroup radioGroup;
    RadioButton radioButton;
    //  Spinner mBranch;
    String rollno;
    Boolean role;
    FirebaseFirestore fStore;
    private static final String TAG = "Register";

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

        Calendar calendar = Calendar.getInstance();
        int year = calendar.get(Calendar.YEAR);


        mName = findViewById(R.id.name);
        mEmail = findViewById(R.id.email);
        mPassword = findViewById(R.id.password);
        mLoginBtn = findViewById(R.id.rloginbutn);
        mSignUpBtn = findViewById(R.id.rsignupbtn);
        mRegistrationID = findViewById(R.id.registrationID);
        mRollno = findViewById(R.id.rollnostudent);
        radioGroup = findViewById(R.id.rdgroup);
```

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

        fAuth = FirebaseAuth.getInstance();
        fStore = FirebaseFirestore.getInstance();

        if(fAuth.getCurrentUser() != null){
            // user already logged in...Directly send to main screen
            Intent i = new Intent(getApplicationContext(), MainActivity.class);
            i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
            startActivity(i);
            finish();
        }

//        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
R.array.Branches, android.R.layout.simple_spinner_item);
//
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
//        mBranch.setAdapter(adapter);
//        mBranch.setOnItemSelectedListener((AdapterView.OnItemSelectedListener) this);

        radioGroup.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup group, int checkedId) {
                radioButton = findViewById(checkedId);
                String roletext = radioButton.getText().toString();
                if (roletext.equals("Student")) {
                    mRollno.setVisibility(View.VISIBLE);
                    role = true;


                } else if(roletext.equals("Teacher")){
                    mRollno.setVisibility(View.INVISIBLE);
                    role = false;
                }
            }
        });
        mLoginBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // user has already an account...Start login activity
                Intent i = new Intent(getApplicationContext(), Login.class);
                i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                startActivity(i);
                finish();
```

```
        }
    });

    mSignUpBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String email = mEmail.getText().toString().trim();
            String pass = mPassword.getText().toString().trim();
            String name = mName.getText().toString();
            String regID = mRegistrationID.getText().toString().toUpperCase();
            String branch = mBranch.getText().toString();

            if(role){
                rollno = mRollno.getText().toString();
            }
            //     String branch = mBranch.getItemAtPosition(i).toString();

            if(TextUtils.isEmpty(email)){
                mEmail.setError("Email is required.");
                return;
            }
            if(TextUtils.isEmpty(pass)){
                mPassword.setError("Password is required.");
                return;
            }
            if(pass.length() < 5){
                mPassword.setError("Password must be >= 5 characters");
                return;
            }
            if(regID.length() != 11){
                mRegistrationID.setError("Wrong format");
                return;
            }

            progressBar.setVisibility(View.VISIBLE);

            // register user in firebase
            fAuth.createUserWithEmailAndPassword(email,
pass).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if(task.isSuccessful()){
                        // AUTHENTICATION SUCCESSFUL
                        Toast.makeText(Register.this, "Account created",
Toast.LENGTH_SHORT).show();
                        // new account created!
                        Intent intent = new Intent(getApplicationContext(), MyDomain.class);
```

```
                    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                    intent.putExtra("activity", TAG);
                    startActivity(intent);

                    DocumentReference docR_c2k = fStore.document("year/"+year+"-
"+(year+1)+"/IDS/"+fAuth.getCurrentUser().getUid());
                    Map<String, Object> idandc2k = new HashMap<>();
                    idandc2k.put("RegID",regID);
                    docR_c2k.set(idandc2k);

                    // ADD DATA TO FIRESTORE
                    DocumentReference docR = fStore.document("year/"+year+"-
"+(year+1)+"/Users/"+regID);

                    //     DocumentReference docref =
fStore.collection("Year").document(year+"-"+(year+1)).collection("Users").document(regID);
                    //     DocumentReference docr =
fStore.collection("Users").document(regID);
                    Map<String, Object> datatosave = new HashMap<>();

                    datatosave.put("Name", name);
                    if(role){
                       datatosave.put("Role", true);
                       datatosave.put("RollNo", rollno);
                    } else{
                       datatosave.put("Role", false);
                       datatosave.put("RollNo", "N.A.");
                    }
                    datatosave.put("Email", email);
                    datatosave.put("Branch", branch);
                    datatosave.put("Password", pass);
                    datatosave.put("GroupID", "N.A.");
                    datatosave.put("PhoneNo", "N.A.");
                    datatosave.put("Linkedin", "N.A.");
                    datatosave.put("Github", "N.A.");
                    datatosave.put("Resume", "N.A.");

                    datatosave.put("RegistrationID", regID);
                    docR.set(datatosave).addOnCompleteListener(new
OnCompleteListener<Void>() {
                       @Override
                       public void onComplete(@NonNull Task<Void> task) {
                          if(task.isSuccessful()){
                             Log.d("Roll", "rollno: "+ rollno);
                             Log.d("Hooray", "Task successful");
                          } else{
                             Log.d("Error", "There was an error!");
```

```
                        }
                    }
                });
            }else{
                // AUTHENTICATION UNSUCCESSFUL
                Toast.makeText(Register.this, "Error..! " +
task.getException().getMessage(),Toast.LENGTH_SHORT).show();
            }
        }
    });
  }
});
}

@Override
public void onBackPressed() {

}


}
```

DomainsAdapter.java

```
package com.example.teproject;

import android.content.Context;
import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseExpandableListAdapter;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.core.content.ContextCompat;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
```

```
import com.google.firebase.firestore.QueryDocumentSnapshot;
import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class DomainsAdapter extends BaseExpandableListAdapter {

    private int year;
    private FirebaseFirestore db;
    FirebaseAuth fAuth;
    private CollectionReference domainsRef;
    private CollectionReference userdomainsRef;

    private static final String TAG = "DomainsAdapter";

    // Sample data set.  children[i] contains the children (String[]) for groups[i].

    // We first get the complete list of domains and subdomains
    public List<String> groups = new ArrayList<>();
    public ArrayList<ArrayList<String>> children = new ArrayList<ArrayList<String>>();

    // There will contain the actual user data
    public List<String> usergroups = new ArrayList<>();
    public ArrayList<ArrayList<String>> userchildren = new ArrayList<ArrayList<String>>();

    private Context context;

    public ArrayList<ArrayList<Integer>> check_states;
    public HashMap<String, Integer> check_groups;

    static boolean flag = false;

    //Constructors
    public DomainsAdapter() {
    }

    public DomainsAdapter(Context c, List<String> groups, ArrayList<ArrayList<String>>
children,
                          List<String> usergroups, ArrayList<ArrayList<String>> userchildren,
                          ArrayList<ArrayList<Integer>> check_states, HashMap<String, Integer>
check_groups) {
        // set the context
        this.context = c;
        this.groups = groups;
```

```
    this.children = children;
    this.usergroups = usergroups;
    this.userchildren = userchildren;
    this.check_states = check_states;
    this.check_groups = check_groups;

}


private void setStates(int i, ArrayList<String> s1) {
    for(int j=0;j<this.groups.size();j++) {
        for(int k=0;k<this.children.get(j).size();k++) {
            String temp = this.children.get(j).get(k);
            if(temp.equals(s1.get(i))) {
                check_states.get(j).set(k, 1);
            }
        }
    }
}



@Override
public int getGroupCount() {
    return groups.size();
}

@Override
public int getChildrenCount(int groupPosition) {
    return children.get(groupPosition).size();
}

@Override
public Object getGroup(int groupPosition) {
    return groups.get(groupPosition);
}

@Override
public Object getChild(int groupPosition, int childPosition) {
    return children.get(groupPosition).get(childPosition);
}

@Override
public long getGroupId(int groupPosition) {
    return groupPosition;
}

@Override
```

```
    public long getChildId(int groupPosition, int childPosition) {
        return childPosition;
    }

    @Override
    public boolean hasStableIds() {
        return false;
    }

    @Override
    public View getGroupView(int groupPosition, boolean isExpanded, View convertView,
ViewGroup parent) {
        View grid;

        if (convertView == null) {
            grid = new View(context);
            LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            grid = inflater.inflate(R.layout.specialty_header, parent, false);
        } else {
            grid = (View) convertView;
        }

        TextView header = (TextView) grid.findViewById(R.id.specialty_header);
        String heading = getGroup(groupPosition).toString();
        header.setText(heading);
        int color;
        if(check_groups.get(heading) == 0)
            color = ContextCompat.getColor(context,R.color.faded_gray);
        else {
            color = ContextCompat.getColor(context, R.color.black);
        }
        header.setTextColor(color);
        return grid;
    }

    @Override
    public View getChildView(int groupPosition, int childPosition, boolean isLastChild, View
convertView, ViewGroup parent) {
        View grid;


        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        grid = inflater.inflate(R.layout.specialty_list_item, parent, false);

        final int grpPos = groupPosition;
        final int childPos = childPosition;
```

```
        TextView header = (TextView) grid.findViewById(R.id.title);
        header.setText(getChild(groupPosition, childPosition).toString());
        final ImageView tick = grid.findViewById(R.id.image_check);
        int color;
        if (check_states.get(grpPos).get(childPos) == 1) {
            tick.setImageResource(R.drawable.ic_checkbox_filled_foreground);
            color = ContextCompat.getColor(context, R.color.black);
        }
        else {
            tick.setImageResource(R.drawable.ic_checkbox_empty_foreground);
            color = ContextCompat.getColor(context,R.color.faded_gray);
        }
        header.setTextColor(color);
        grid.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int x = check_states.get(grpPos).get(childPos);
                int color;
                if(x == 0) {
                    check_states.get(grpPos).set(childPos, 1);
                    tick.setImageResource(R.drawable.ic_checkbox_filled_foreground);
                    color = ContextCompat.getColor(context, R.color.black);
                }
                else {
                    check_states.get(grpPos).set(childPos, 0);
                    tick.setImageResource(R.drawable.ic_checkbox_empty_foreground);
                    color = ContextCompat.getColor(context,R.color.faded_gray);
                }
                header.setTextColor(color);

                // now, here check if all the children of parent are off or not. If off dim the parent
else make it bold

                ArrayList<Integer> temp = new ArrayList<>();
                temp = check_states.get(groupPosition);
                for(int p=0;p<temp.size();p++) {
                    if(temp.get(p) == 1) {
                        flag = true;
                        break;
                    }
                }
                String message = "";
                if(flag == false) {
                    // toggle parent
                    String heading = (String) getGroup(groupPosition);
                    check_groups.put(heading, 0);
                    for(Map.Entry m: check_groups.entrySet()) {
                        message += m.getKey().toString() + " " + m.getValue().toString() + "\n";
```

```
                }
//                Toast.makeText(context, message, Toast.LENGTH_SHORT).show();
            }
            else {
                // keep the parent highlighted
                String heading = (String) getGroup(groupPosition);
                check_groups.put(heading, 1);
                for(Map.Entry m: check_groups.entrySet()) {
                    message += m.getKey().toString() + " " + m.getValue().toString() + "\n";
                }
//                Toast.makeText(context, message, Toast.LENGTH_SHORT).show();
                flag = false;
            }

        }
    });


    return grid;
  }

  public void toggleParent() {

  }

  @Override
  public boolean isChildSelectable(int groupPosition, int childPosition) {
    return true;
  }
}
```

EditGroup.java

```
package com.example.teproject;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.Toast;

import androidx.annotation.NonNull;
```

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

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FieldValue;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.SetOptions;

import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;

import static com.google.firebase.firestore.FieldValue.arrayUnion;


public class EditGroup extends AppCompatActivity {

    Calendar calendar = Calendar.getInstance();
    int year = calendar.get(Calendar.YEAR);

    private static final String TAG = "MainActivity";
    private LinearLayout parentLinearLayout;
    EditText text;
    FirebaseFirestore fStore;
    FirebaseAuth fAuth;
    DocumentReference docR, docR2;
    String RegID, GroupID;
    Button add,submit;
    ImageView delete;

    String problem_statement="";
    private Toolbar mToolbar;


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

        mToolbar = findViewById(R.id.back_toolbar);
        mToolbar.setNavigationIcon(R.drawable.ic_arrow);
        mToolbar.setNavigationOnClickListener(new View.OnClickListener() {
```

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

            // Your code
            Intent groupIntent = new Intent(getApplicationContext(), GroupProfile.class);
            groupIntent.putExtra("activity", TAG);
            startActivity(groupIntent);
            finish();
        }
    });


    parentLinearLayout=(LinearLayout) findViewById(R.id.parent_linear_layout);
    text = findViewById(R.id.problem_statement_text);
    add = findViewById(R.id.button_add);
    delete = findViewById(R.id.delete_button);
    submit = findViewById(R.id.button_submit);

    fAuth = FirebaseAuth.getInstance();
    fStore = FirebaseFirestore.getInstance();

    docR = fStore.document("year/"+year+"-
"+(year+1)+"/IDS/"+fAuth.getCurrentUser().getUid());

    docR.get().addOnSuccessListener(this, new
OnSuccessListener<DocumentSnapshot>() {
        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            if(documentSnapshot.exists()){
                RegID = documentSnapshot.getString("RegID");
                Log.d("ch", RegID);
                getGroupID();
            }

        }
    });



}



public void getDetails(){
    docR = fStore.document("year/"+year+"- "+(year+1)+"/Groups/"+GroupID);
    docR.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
        @Override
        public void onComplete(@NonNull Task<DocumentSnapshot> task) {
```

```
            DocumentSnapshot document = task.getResult();
            if (document.exists()) {
                problem_statement = document.getString("ProblemStatement");
                if(problem_statement != null){
                    text.setText(problem_statement);
                }
                Map<String, Object> map1 = document.getData();
                for (Map.Entry<String, Object> entry : map1.entrySet()) {
                    if (entry.getKey().equals("TechStack")) {
                        String tech = (entry.getValue().toString());
                        tech = tech.substring(1, tech.length()-1);
                        String[] techh = tech.split(", ", -2);

                        for (String a : techh){
                            if (!a.equals("null")){
                                addTech(a);
                            }
                        }

                    }
                }

            }
        }
    });
  }

  public void addTech(String tech){
      LayoutInflater inflater=(LayoutInflater)
getSystemService(Context.LAYOUT_INFLATER_SERVICE);
      final View rowView=inflater.inflate(R.layout.field, null);
      EditText text = rowView.findViewById(R.id.tech_edit_text);
      text.setText(tech);
      // Add the new row before the add field button.
      parentLinearLayout.addView(rowView, parentLinearLayout.getChildCount() - 1);

  }




  public void getGroupID(){
      docR2 = fStore.document("year/" + year + "- " + (year + 1) + "/Users/" + RegID);
      docR2.get().addOnSuccessListener(this, new
OnSuccessListener<DocumentSnapshot>() {

          @Override
```

```
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                if (documentSnapshot.exists()) {
                    GroupID = documentSnapshot.getString("GroupID");
                    Log.d("c", GroupID);
                    getDetails();

                }
            }
        });
    }


    public void onAddField(View v) {
        LayoutInflater inflater=(LayoutInflater)
getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        final View rowView=inflater.inflate(R.layout.field, null);
        // Add the new row before the add field button.
        parentLinearLayout.addView(rowView, parentLinearLayout.getChildCount() - 1);
    }



    public void onDelete(View v) {
        parentLinearLayout.removeView((View) v.getParent());

    }


    public void onSubmit(View v){

        docR = fStore.document("year/"+year+"- "+(year+1)+"/Groups/"+GroupID);

        Map<String, Object> datatosave = new HashMap<>();
        String[] techStack = new String[20];

        int childcount = parentLinearLayout.getChildCount();

        String problem_statement = text.getText().toString().trim();
        if (!problem_statement.isEmpty())
            datatosave.put("ProblemStatement", problem_statement);


        for (int i=0; i < childcount; i++){
            View view = parentLinearLayout.getChildAt(i);
            // do whatever you would want to do with this View
            EditText text = view.findViewById(R.id.tech_edit_text);
            String tech= (text.getText().toString().trim());
            if (!(tech.isEmpty())){
```

```
            techStack[i] = tech;
        }


    }

    datatosave.put("TechStack", arrayUnion(techStack));


    docR.set(datatosave, SetOptions.merge()).addOnCompleteListener(new
OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            if(task.isSuccessful()){
                Toast.makeText(EditGroup.this, "Saved", Toast.LENGTH_SHORT).show();
            } else{
                Toast.makeText(EditGroup.this, "Try Again", Toast.LENGTH_SHORT).show();
            }
        }
    });

  }

}
```

GroupProfile.java

```
package com.example.teproject;

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

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
```

```
import com.google.firebase.firestore.SetOptions;
import com.google.gson.Gson;

import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;

import static com.google.firebase.firestore.FieldValue.arrayRemove;


public class GroupProfile extends AppCompatActivity {

    FirebaseFirestore fStore;
    FirebaseAuth fAuth;
    String RegID, GroupID = "";
    TextView group_id, problem_statement, mentor_id, member1, techStack;
    Button mLeaveGrpbtn, edit_group;
    DocumentReference docR, docR_2,docR_3;
    private boolean mRole;
    private int year;
    private Toolbar mToolbar;

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

        mToolbar = findViewById(R.id.back_toolbar);
        mToolbar.setNavigationIcon(R.drawable.ic_arrow);
        mToolbar.setNavigationOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View view) {

                // Your code
                finish();
            }
        });

        Calendar calendar = Calendar.getInstance();
        year = calendar.get(Calendar.YEAR);

        group_id = findViewById(R.id.groupid);
        problem_statement = findViewById(R.id.problem_statement_id);
        mentor_id = findViewById(R.id.mentor_id);
        member1 = findViewById(R.id.member1);
        techStack = findViewById(R.id.tech_stack_id);
        edit_group = findViewById(R.id.edit_group);
```

```
        fAuth = FirebaseAuth.getInstance();
        fStore = FirebaseFirestore.getInstance();

        // getting the caller activity/fragment
        Intent groupIntent = getIntent();
        String caller = groupIntent.getStringExtra("activity");
        Log.d("Caller", caller);

        if (caller.equals("MainActivity")) {
            docR = fStore.document("year/"+year+"-
"+(year+1)+"/IDS/"+fAuth.getCurrentUser().getUid());

            docR.get().addOnSuccessListener(this, new
OnSuccessListener<DocumentSnapshot>() {
                @Override
                public void onSuccess(DocumentSnapshot documentSnapshot) {
                    if(documentSnapshot.exists()){
                        RegID = documentSnapshot.getString("RegID");
                        tp(year);
                    }
                }
            });
        } else{
            // extract other details from intent
            Gson gson = new Gson();
            GroupsOverview groupsOverview =
gson.fromJson(groupIntent.getStringExtra("groupsOverView"), GroupsOverview.class);
            String groupId = groupIntent.getStringExtra("groupId");

            // hide the edit button
            edit_group.setVisibility(View.GONE);
            // now setting the values of components:
            group_id.setText(groupId);
            if(groupsOverview.getMentorID() != null)
                mentor_id.setText(groupsOverview.getMentorID());
            if(groupsOverview.getProblemStatement() != null)
                problem_statement.setText(groupsOverview.getProblemStatement());
            if(groupsOverview.getMembers() != null) {
                String mem = (groupsOverview.getMembers().toString());
                mem = mem.substring(1, mem.length()-1);
                String[] members = mem.split(", ", -2);

                for (String a : members){
                    //  member1.append(a+"\n");
                    a.trim();
                    members(a);
                }
```

```
            }
            if(groupsOverview.getTechStack() != null) {
                String tech = (groupsOverview.getTechStack().toString());
                tech = tech.substring(1, tech.length()-1);
                String[] members = tech.split(", ", -2);

                for (String a : members){
                    if (!a.equals("null")){
                        techStack.append(a+"\n");
                    }

                }
            }
        }


        edit_group.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getApplicationContext(), EditGroup.class));
            }
        });


    }


    void tp(int year){
        docR_2 = fStore.document("year/"+year+"- "+(year+1)+"/Users/"+RegID);
        Log.d("check1", " in tp check1 - doc");
        docR_2.get().addOnSuccessListener(this, new
OnSuccessListener<DocumentSnapshot>() {

            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                if(documentSnapshot.exists()) {
                    Log.d("check2", " in tp check2 - exists");
                    GroupID = documentSnapshot.getString("GroupID");
                    mRole  = documentSnapshot.getBoolean("Role");
                    Log.d("TAG", "GroupID"+GroupID);

                    if(!GroupID.isEmpty()){
                        group(year, GroupID);
                    }
                }
            }
        });
    }
```

```
    void members(String mem){
        docR_2 = fStore.document("year/"+year+"- "+(year+1)+"/Users/"+mem);
        docR_2.get().addOnSuccessListener(this, new
OnSuccessListener<DocumentSnapshot>() {

            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                String name = documentSnapshot.getString("Name").toString();
                member1.append(mem+ " - "+ name+ "\n");
            }
        });

    }

    void getMentor(String mem){
        docR_2 = fStore.document("year/"+year+"- "+(year+1)+"/Users/"+mem);
        docR_2.get().addOnSuccessListener(this, new
OnSuccessListener<DocumentSnapshot>() {

            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                String name = documentSnapshot.getString("Name").toString();
                mentor_id.setText(mem+ " - "+ name+ "\n");
            }
        });

    }

    void group(int year, String GroupID){
        docR_3 = fStore.document("year/"+year+"- "+(year+1)+"/Groups/"+GroupID);
        Log.d("check1", "in group check1 - doc");
        docR_3.get().addOnSuccessListener(this, new
OnSuccessListener<DocumentSnapshot>() {

            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                if(documentSnapshot.exists()) {
                    Log.d("check2", " in group check2 - exists");
                    group_id.setText(GroupID);
                        String mentor = (documentSnapshot.getString("MentorID"));
                    if (mentor != null)
                        getMentor(mentor);

problem_statement.setText(documentSnapshot.getString("ProblemStatement"));

                    Map<String, Object> map = documentSnapshot.getData();
                    for (Map.Entry<String, Object> entry : map.entrySet()) {
```

```
                    if (entry.getKey().equals("Members")) {
                        String mem = (entry.getValue().toString());
                        mem = mem.substring(1, mem.length()-1);
                        String[] members = mem.split(", ", -2);

                        for (String a : members){
                          //  member1.append(a+"\n");
                           a.trim();
                           members(a);
                        }
                    }
                }

                Map<String, Object> map1 = documentSnapshot.getData();
                for (Map.Entry<String, Object> entry : map1.entrySet()) {
                    if (entry.getKey().equals("TechStack")) {
                        String tech = (entry.getValue().toString());
                        tech = tech.substring(1, tech.length()-1);
                        String[] members = tech.split(", ", -2);

                        for (String a : members){
                            if (!a.equals("null")){
                                techStack.append(a+"\n");
                            }

                        }

                    }
                }


            }
        }
    });
  }

}
```

GroupsOverview.java

```
package com.example.teproject;

import java.util.List;

public class GroupsOverview {
   private String ProblemStatement;
   private String GroupID;
   private String MentorID;
```

```
private List<String> Members;
private List<String> TechStack;

public GroupsOverview() {
}

public GroupsOverview(String groupID) {
    GroupID = groupID;
}

public String getGroupID() {
    return GroupID;
}

public void setGroupID(String groupID) {
    GroupID = groupID;
}

public String getProblemStatement() {
    return ProblemStatement;
}

public void setProblemStatement(String problemStatement) {
    ProblemStatement = problemStatement;
}

public String getMentorID() {
    return MentorID;
}

public void setMentorID(String mentorID) {
    MentorID = mentorID;
}

public List<String> getMembers() {
    return Members;
}

public void setMembers(List<String> members) {
    Members = members;
}

public List<String> getTechStack() {
    return TechStack;
}

public void setTechStack(List<String> techStack) {
    TechStack = techStack;
```

```
    }
}
```

HomeFragment.java

```
package com.example.teproject;

import android.content.Intent;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.Calendar;
import java.util.Map;


public class HomeFragment extends Fragment {
    FirebaseFirestore fStore;
    FirebaseAuth fAuth;
    String RegID, GroupID = "";
    TextView group_id, problem_statement, mentor_id, member1, techStack;
    Button mLeaveGrpbtn, edit_group;
    DocumentReference docR, docR_2,docR_3;
    private boolean mRole;
    private int year;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.fragment_home, container, false);

        return v;
```

```
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState)
{
        super.onViewCreated(view, savedInstanceState);
        Calendar calendar = Calendar.getInstance();
        year = calendar.get(Calendar.YEAR);

        group_id = view.findViewById(R.id.groupid);
        problem_statement = view.findViewById(R.id.problem_statement_id);
        mentor_id = view.findViewById(R.id.mentor_id);
        member1 = view.findViewById(R.id.member1);
        techStack = view.findViewById(R.id.tech_stack_id);
        edit_group = view.findViewById(R.id.edit_group);

        fAuth = FirebaseAuth.getInstance();
        fStore = FirebaseFirestore.getInstance();

        edit_group.setVisibility(View.GONE);
        docR = fStore.document("year/"+year+"-
"+(year+1)+"/IDS/"+fAuth.getCurrentUser().getUid());

        docR.get().addOnSuccessListener(getActivity(), new
OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                if(documentSnapshot.exists()){
                    RegID = documentSnapshot.getString("RegID");
                    tp(year);
                }
            }
        });
    }

    void tp(int year){
        docR_2 = fStore.document("year/"+year+"- "+(year+1)+"/Users/"+RegID);
        Log.d("check1", " in tp check1 - doc");
        docR_2.get().addOnSuccessListener(getActivity(), new
OnSuccessListener<DocumentSnapshot>() {

            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                if(documentSnapshot.exists()) {
                    Log.d("check2", " in tp check2 - exists");
                    GroupID = documentSnapshot.getString("GroupID");
                    mRole  = documentSnapshot.getBoolean("Role");
                    Log.d("TAG", "GroupID"+GroupID);
```

```
                if(!GroupID.isEmpty()){
                    group(year, GroupID);
                }
            }
        }
    });
}

void members(String mem){
    docR_2 = fStore.document("year/"+year+"- "+(year+1)+"/Users/"+mem);
    docR_2.get().addOnSuccessListener(getActivity(), new
OnSuccessListener<DocumentSnapshot>() {

        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            String name = documentSnapshot.getString("Name").toString();
            member1.append(mem+ " - "+ name+ "\n");
        }
    });

}

void getMentor(String mem){
    docR_2 = fStore.document("year/"+year+"- "+(year+1)+"/Users/"+mem);
    docR_2.get().addOnSuccessListener(getActivity(), new
OnSuccessListener<DocumentSnapshot>() {

        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            String name = documentSnapshot.getString("Name").toString();
            mentor_id.setText(mem+ " - "+ name+ "\n");
        }
    });

}

void group(int year, String GroupID){
    docR_3 = fStore.document("year/"+year+"- "+(year+1)+"/Groups/"+GroupID);
    Log.d("check1", "in group check1 - doc");
    docR_3.get().addOnSuccessListener(getActivity(), new
OnSuccessListener<DocumentSnapshot>() {

        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            if(documentSnapshot.exists()) {
                Log.d("check2", " in group check2 - exists");
                group_id.setText(GroupID);
```

```
                //        mLeaveGrpbtn.setVisibility(View.VISIBLE);
                String mentor = (documentSnapshot.getString("MentorID"));
                if (mentor != null)
                    getMentor(mentor);

problem_statement.setText(documentSnapshot.getString("ProblemStatement"));

                Map<String, Object> map = documentSnapshot.getData();
                for (Map.Entry<String, Object> entry : map.entrySet()) {
                    if (entry.getKey().equals("Members")) {
                        String mem = (entry.getValue().toString());
                        mem = mem.substring(1, mem.length()-1);
                        String[] members = mem.split(", ", -2);

                        for (String a : members){
                           // member1.append(a+"\n");
                            a.trim();
                            members(a);
                        }
                    }
                }

                Map<String, Object> map1 = documentSnapshot.getData();
                for (Map.Entry<String, Object> entry : map1.entrySet()) {
                    if (entry.getKey().equals("TechStack")) {
                        String tech = (entry.getValue().toString());
                        tech = tech.substring(1, tech.length()-1);
                        String[] members = tech.split(", ", -2);

                        for (String a : members){
                            if (!a.equals("null")){
                                techStack.append(a+"\n");
                            }

                        }

                    }
                }


            }
        }
    });
  }
}
```

ListGroupAdapter.java

```
package com.example.teproject;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ExpandableListView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.firebase.ui.firestore.FirestoreRecyclerAdapter;
import com.firebase.ui.firestore.FirestoreRecyclerOptions;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;

public class ListGroupsAdapter extends FirestoreRecyclerAdapter<GroupsOverview,
ListGroupsAdapter.GroupHolder> {

    private OnGroupItemClickListener listener;
    public ListGroupsAdapter(@NonNull FirestoreRecyclerOptions<GroupsOverview>
options) {
        super(options);
    }

    @Override
    protected void onBindViewHolder(@NonNull GroupHolder holder, int position, @NonNull
GroupsOverview model) {
        if(model == null) {
            Log.d("lOGGING!!", "Model is null");
        }
        DocumentSnapshot documentSnapshot = getSnapshots().getSnapshot(position);
        String currGroupId = documentSnapshot.getId();

        holder.groupId.setText("Group Id: "+currGroupId);
        if (model.getProblemStatement() != null) {
            String probStatement = model.getProblemStatement();
            if(probStatement.length() > 22) {
                probStatement = probStatement.substring(0, 21);
                probStatement += "...";
            }
            holder.problemStatement.setText("Problem Statement: "+probStatement);
        }
    }

    @NonNull
```

```
    @Override
    public GroupHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.group_list_item,
parent, false);
        return new GroupHolder(v);
    }

    class GroupHolder extends RecyclerView.ViewHolder {
        TextView problemStatement, groupId;
        public GroupHolder(@NonNull View itemView) {
            super(itemView);
            groupId = itemView.findViewById(R.id.group_name);
            problemStatement = itemView.findViewById(R.id.problem_statement);
            itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    int position = getAdapterPosition();
                    if (position != RecyclerView.NO_POSITION && listener != null) {
                        listener.onGrpItemClick(getSnapshots().getSnapshot(position), position);
                    }
                }
            });
        }
    }

    public interface OnGroupItemClickListener {
        void onGrpItemClick(DocumentSnapshot documentSnapshot, int position);
    }
    public void setOnGroupItemClickListener(OnGroupItemClickListener listener) {
        this.listener = listener;
    }
}
```

ListUsersAdapter.java

```
package com.example.teproject;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.firebase.ui.firestore.FirestoreRecyclerAdapter;
import com.firebase.ui.firestore.FirestoreRecyclerOptions;
```

```
import com.google.firebase.firestore.DocumentSnapshot;

public class ListUsersAdapter extends FirestoreRecyclerAdapter<UserOverview,
ListUsersAdapter.UserHolder> {

    private OnItemClickListener listener;

    public ListUsersAdapter(@NonNull FirestoreRecyclerOptions<UserOverview> options) {
        super(options);
    }

    @Override
    protected void onBindViewHolder(@NonNull UserHolder holder, int position, @NonNull
UserOverview model) {
        // holder.itemView.component for accessing
        // what we have to put in each layout in our CardView

        holder.txtName.setText(model.getName());
        holder.txtBranch.setText(model.getBranch());
        holder.txtEmail.setText(model.getEmail());
    }

    @NonNull
    @Override
    public UserHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.person_list_item,
parent, false);
        return new UserHolder(view);
    }

    class UserHolder extends RecyclerView.ViewHolder {

        TextView txtName, txtBranch, txtEmail;
        public UserHolder(@NonNull View itemView) {
            super(itemView);
            txtName = itemView.findViewById(R.id.usr_full_name);
            txtBranch = itemView.findViewById(R.id.usr_branch_name);
            txtEmail = itemView.findViewById(R.id.usr_email);

            itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    int position = getAdapterPosition();
                    if(position != RecyclerView.NO_POSITION && listener != null) {
                        listener.onItemClick(getSnapshots().getSnapshot(position), position);
                    }
                }
            });
```

```
        }
    }

    public interface OnItemClickListener {
        void onItemClick(DocumentSnapshot documentSnapshot, int position);
    }
    public void setOnItemClickListener(OnItemClickListener listener) {
        this.listener = listener;

    }
}
```

ListUsersFragment.java

```
package com.example.teproject;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.os.Parcelable;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Adapter;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListAdapter;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.firebase.ui.firestore.FirestoreRecyclerAdapter;
import com.firebase.ui.firestore.FirestoreRecyclerOptions;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.firestore.CollectionReference;
```

```
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.Query;
import com.google.firebase.firestore.QuerySnapshot;
import com.google.gson.Gson;

import org.w3c.dom.Text;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;

public class ListUsersFragment extends Fragment {
    private static final String TAG = "ListUsersFragment";
    private static final String CALLING_TAG = "ListUserFragment";

    private View studentsFragmentsView;
    private RecyclerView recyclerView;
    private TextView txt_title;

    private int year;
    private FirebaseFirestore db;
    private CollectionReference userdomainsRef;
    private CollectionReference domainsRef;
    private EditText txtSearch;

    private ListUsersAdapter adapter;

    private ArrayList<String> listDomains;
    private Button btnFilter;
    private int checkedInitialItems;
    private int backUp;
    private String arrDomains[];

    String tabName;
    public ListUsersFragment(String tabName) {
        this.tabName = tabName;
    }

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        studentsFragmentsView = inflater.inflate(R.layout.fragment_list_users, container, false);

        Calendar calendar = Calendar.getInstance();
        year = calendar.get(Calendar.YEAR);
        db = FirebaseFirestore.getInstance();
```

```
        userdomainsRef = db.collection("year/"+year+"- "+(year+1)+"/Users");
        domainsRef = db.collection("year/"+year+"- "+(year+1)+"/Domains");

        btnFilter = studentsFragmentsView.findViewById(R.id.btn_filter);
        listDomains = new ArrayList<>();
        readData(new FirestoreCallback() {
            @Override
            public void onCallBack(List<String> list) {
                listDomains = new ArrayList<>();
                listDomains.add("All");
                for (int x=0;x<list.size();x++) {
                    listDomains.add(list.get(x));
                }

                arrDomains = new String[listDomains.size()];
                for (int x=0;x<listDomains.size();x++) {
                    arrDomains[x] = listDomains.get(x);
                }
                checkedInitialItems = 0;
                backUp = 0;

//              Toast.makeText(getContext(), listDomains.toString(),
Toast.LENGTH_SHORT).show();
                Toast.makeText(getContext(), "Data Loaded!", Toast.LENGTH_SHORT).show();
                ArrayList<String> selectedItems = new ArrayList<>();

                btnFilter.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        AlertDialog.Builder builder = new AlertDialog.Builder(getContext());
                        builder.setTitle("Select Domains")
                                .setSingleChoiceItems(arrDomains, checkedInitialItems, new
DialogInterface.OnClickListener() {
                                    @Override
                                    public void onClick(DialogInterface dialogInterface, int i) {
                                        checkedInitialItems = i;
                                    }
                                });
                        builder.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialogInterface, int which) {

                                backUp = checkedInitialItems;

                                String finalSelected = listDomains.get(checkedInitialItems);
                                Log.d("Finally... ", finalSelected);

                                boolean tflag = false;
```

```
                    if (checkedInitialItems == 0) {
                        tflag = false;
                    } else
                        tflag = true;

                    if (tflag == false) {
                        // all entries are false....do not do anything
                        Query query = userdomainsRef;
                        if(tabName.equals("Students")) {
                            query = userdomainsRef.whereEqualTo("Role", true);
                        } else {
                            query = userdomainsRef.whereEqualTo("Role", false);
                            txt_title.setText("Find your mentors");
                        }
                        FirestoreRecyclerOptions<UserOverview> options = new
FirestoreRecyclerOptions.Builder<UserOverview>()
                                .setQuery(query, UserOverview.class)
                                .build();
                        adapter.updateOptions(options);

                    }
                    else {
                        Query q2;
                        if(tabName.equals("Students")) {
                            q2 = userdomainsRef.whereEqualTo("Role", true);
                            String str = "Domains.";
                            str += finalSelected;
                            q2 = q2.whereNotEqualTo(str, "");
                        } else {
                            q2 = userdomainsRef.whereEqualTo("Role", false);
                            String str = "Domains.";
                            str += finalSelected;
                            q2 = q2.whereNotEqualTo(str, "");
                            txt_title.setText("FIND YOUR MENTORS");
                        }
                        FirestoreRecyclerOptions<UserOverview> options2 = new
FirestoreRecyclerOptions.Builder<UserOverview>()
                                .setQuery(q2, UserOverview.class)
                                .build();
                        adapter.updateOptions(options2);
                    }
                    // ------------------------------------

                }
            });
            builder.setNeutralButton("Cancel", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {
```

```
                        // do something
                        // revert the checked array back to backup array
                        checkedInitialItems = backUp;
                    }
                });
                AlertDialog dialog = builder.create();
                dialog.show();
            }
        });

    }
});


    txtSearch = studentsFragmentsView.findViewById(R.id.txt_search_users);
    recyclerView = studentsFragmentsView.findViewById(R.id.students_recycler_view);
    recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));
    return studentsFragmentsView;
}

private void readData(FirestoreCallback firestoreCallback) {
    domainsRef.get().addOnCompleteListener(new
OnCompleteListener<QuerySnapshot>() {
        @Override
        public void onComplete(@NonNull Task<QuerySnapshot> task) {
            if (task.isSuccessful()) {
                for (DocumentSnapshot documentSnapshot: task.getResult()) {
                    String currDomain = (String) documentSnapshot.get("name");
                    Log.d("Current: ", currDomain);
                    listDomains.add(currDomain);
                }
                firestoreCallback.onCallBack(listDomains);
            } else
                Log.d(TAG, "Error fetching data!");
        }
    });
}
private interface FirestoreCallback {
    void onCallBack(List<String> list);
}

@Override
public void onStart() {
    super.onStart();
    Query query;
    txt_title = studentsFragmentsView.findViewById(R.id.list_heading);
    if(tabName.equals("Students")) {
        query = userdomainsRef.whereEqualTo("Role", true);
```

```
        } else {
            query = userdomainsRef.whereEqualTo("Role", false);
            txt_title.setText("FIND YOUR MENTORS");
        }
        FirestoreRecyclerOptions<UserOverview> options = new
FirestoreRecyclerOptions.Builder<UserOverview>()
                .setQuery(query, UserOverview.class)
                .build();

        adapter = new ListUsersAdapter(options);

        recyclerView.setAdapter(adapter);
        adapter.setOnItemClickListener(new ListUsersAdapter.OnItemClickListener() {
            @Override
            public void onItemClick(DocumentSnapshot documentSnapshot, int position) {
                UserOverview userOverview = documentSnapshot.toObject(UserOverview.class);
                String name = userOverview.getName();
//                Toast.makeText(getContext(), "Position: "+position+" Name: "+name,
Toast.LENGTH_SHORT).show();
                Intent profileIntent = new Intent(getContext(), myprofile.class);

                // passing this to identify caller activity/fragment
                profileIntent.putExtra("caller", CALLING_TAG);

                // serialising the data because custom objects cannot be directly passed!
                Gson gson = new Gson();
                String currUser = gson.toJson(userOverview);
                profileIntent.putExtra("userOverview", currUser);
                startActivity(profileIntent);

            }
        });
        adapter.startListening();

        txtSearch.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

            }

            @Override
            public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

            }

            @Override
            public void afterTextChanged(Editable editable) {
                filter(editable.toString());
```

```
            }
        });
    }
    private void filter(String toString) {
        Query query = userdomainsRef;
        if(tabName.equals("Students")) {
            query = userdomainsRef.whereEqualTo("Role", true);
        } else {
            query = userdomainsRef.whereEqualTo("Role", false);
            txt_title.setText("FIND YOUR MENTORS");
        }
        FirestoreRecyclerOptions<UserOverview> options = new
FirestoreRecyclerOptions.Builder<UserOverview>()
                .setQuery(query, UserOverview.class)
                .build();

        Query q2;
        if(tabName.equals("Students")) {
            q2 = userdomainsRef.whereEqualTo("Role", true)
                    .whereGreaterThanOrEqualTo("Name", toString)
                    .whereLessThanOrEqualTo("Name", toString+"\uf8ff");
        } else {
            q2 = userdomainsRef.whereEqualTo("Role", false)
                    .whereGreaterThanOrEqualTo("Name", toString)
                    .whereLessThanOrEqualTo("Name", toString+"\uf8ff");
            txt_title.setText("FIND YOUR MENTORS");
        }
        FirestoreRecyclerOptions<UserOverview> options2 = new
FirestoreRecyclerOptions.Builder<UserOverview>()
                .setQuery(q2, UserOverview.class)
                .build();
        if(toString.equals("")) {
            adapter.updateOptions(options);
        } else
            adapter.updateOptions(options2);
    }

}
```

MyDomain.java

```
package com.example.teproject;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatDelegate;
import androidx.appcompat.widget.Toolbar;
```

```
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ExpandableListView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QueryDocumentSnapshot;
import com.google.firebase.firestore.QuerySnapshot;
import com.google.firebase.firestore.SetOptions;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class MyDomain extends AppCompatActivity {

    ExpandableListView domainsList;
    DomainsAdapter adapter;
    Button btnSave;
    String RegID;
    Toolbar mToolbar;

    private int year;
    FirebaseAuth fAuth;
    private FirebaseFirestore db;
    private CollectionReference domainsRef;
    private CollectionReference userdomainsRef;

    String activity;



    // We first get the complete list of domains and subdomains
```

```
public List<String> groups = new ArrayList<>();
public ArrayList<ArrayList<String>> children = new ArrayList<ArrayList<String>>();

// These will contain the actual user data
public List<String> usergroups = new ArrayList<>();
public ArrayList<ArrayList<String>> userchildren = new ArrayList<ArrayList<String>>();

public ArrayList<ArrayList<Integer>> check_states = new ArrayList<>();
public HashMap<String, Integer> check_groups = new HashMap<>();

private static final String TAG = "MyDomain";



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

    btnSave = findViewById(R.id.btn_save);
    Intent intent = getIntent();
    activity = intent.getStringExtra("activity");
    if(activity.equals("Register")) {
        btnSave.setText("NEXT");

    } else if(activity.equals("joingroup")) {
        btnSave.setText("SAVE");
    } else {
        btnSave.setText("SAVE");
    }

    mToolbar = findViewById(R.id.back_toolbar);
    mToolbar.setNavigationIcon(R.drawable.ic_arrow);
    mToolbar.setNavigationOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View view) {

            // Your code
            finish();
        }
    });

    // setting up connection...
    Calendar calendar = Calendar.getInstance();
    year = calendar.get(Calendar.YEAR);
    db = FirebaseFirestore.getInstance();
    domainsRef = db.collection("year/"+year+"- "+(year+1)+"/Domains");
```

```
        userdomainsRef = db.collection("year/"+year+"- "+(year+1)+"/Users");
        fAuth = FirebaseAuth.getInstance();


        // first get the list of all domains and subdomains
        loadDomains(new FirestoretoreCallback() {
            @Override
            public void callback(List<String> list1, ArrayList<ArrayList<String>> list2) {
                groups = list1;
                children = list2;
                Log.d(TAG, groups.toString());
                Log.d(TAG, children.toString());
            }
        });

    }

    private void loadDomains(FirestoretoreCallback firestoretoreCallback) {
        domainsRef.get().addOnCompleteListener(new
OnCompleteListener<QuerySnapshot>() {
            @Override
            public void onComplete(@NonNull Task<QuerySnapshot> task) {
                if(task.isSuccessful()) {
                    for(DocumentSnapshot documentSnapshot: task.getResult()) {
                        String name = (String) documentSnapshot.get("name");
                        ArrayList<String> subNames = (ArrayList<String>)
documentSnapshot.get("subdomains");
                        groups.add(name);
                        children.add(subNames);
                    }
                    firestoretoreCallback.callback(groups, children);
                    Log.d(TAG, "Task 1 Complete");


                    DocumentReference docR = db.document("year/"+year+"-
"+(year+1)+"/IDS/"+fAuth.getCurrentUser().getUid());
                    docR.get().addOnCompleteListener(new
OnCompleteListener<DocumentSnapshot>() {
                        @Override
                        public void onComplete(@NonNull Task<DocumentSnapshot> task) {
                            String tempId;
                            if(task.isSuccessful()) {
                                tempId = (String) task.getResult().get("RegID");
                                RegID = tempId;
                            }
                            else {
                                Log.d(TAG, "Could not find RegID");
                            }
```

```
                    Log.d(TAG, "Registration ID: "+RegID);
                    Log.d(TAG, "Task 2 completed!");

                    // moving on to find user's domains and subdomains
                    userdomainsRef.whereEqualTo("RegistrationID", RegID)
                            .get()
                            .addOnCompleteListener(new
OnCompleteListener<QuerySnapshot>() {
                                @Override
                                public void onComplete(@NonNull Task<QuerySnapshot> task) {
                                    if(task.isSuccessful()) {
                                        QuerySnapshot querySnapshot = task.getResult();
                                        for(DocumentSnapshot documentSnapshot: querySnapshot) {
                                            Map<String, ArrayList<String>> subDomains = new
HashMap<>();
                                            subDomains = (Map<String, ArrayList<String>>)
documentSnapshot.get("Domains");

                                            if(activity.equals("Register") || subDomains == null) {
                                                //
                                            }
                                            else {
                                                for(Map.Entry m: subDomains.entrySet()) {
                                                    String dname = m.getKey().toString();
                                                    usergroups.add(dname);
                                                    ArrayList<String> snames = (ArrayList<String>)
m.getValue();

                                                    userchildren.add(snames);
                                                }
                                            }

                                        }
                                        Log.d(TAG, "Completed Task 3");
                                        Log.d(TAG, usergroups.toString());
                                        Log.d(TAG, userchildren.toString());

                                        // moving to next task ie initialising boolean array to 0

                                        check_states = new ArrayList<>();
                                        for (int i = 0; i < groups.size(); i++) {
                                            ArrayList<Integer> tmp = new ArrayList<Integer>();
                                            for (int j = 0; j <children.get(i).size(); j++) {
                                                tmp.add(0);
                                            }
                                            check_states.add(tmp);
                                        }
                                        Log.d(TAG, "Completed Task 4");
                                        Log.d(TAG, check_states.toString());
```

```
                                // moving to next!
                                for(int i=0;i<usergroups.size();i++) {
                                    String currGrp = usergroups.get(i);
                                    int idx1 = groups.indexOf(currGrp);
                                    ArrayList<String> target = children.get(idx1);
                                    ArrayList<String> src = userchildren.get(i);
                                    for(int j=0;j<src.size();j++) {
                                        int idx2 = target.indexOf(src.get(j));
                                        check_states.get(idx1).set(idx2, 1);
                                    }
                                }
                                Log.d(TAG, "Modified boolean array");
                                Log.d(TAG, check_states.toString());

                                check_groups = new HashMap<>();
                                for (int i=0;i<groups.size();i++) {
                                    check_groups.put(groups.get(i), 0);
                                }

                                for(int i=0;i<usergroups.size();i++) {
                                    check_groups.put(usergroups.get(i), 1);
                                }

                                Log.d(TAG, "Check Groups done!");
                                Log.d(TAG, check_groups.toString());

                                // we are here now!

                                domainsList = findViewById(R.id.domains_list);
                                adapter = new DomainsAdapter(getApplicationContext(),
groups,
                                        children, usergroups, userchildren, check_states,
check_groups);
                                domainsList.setAdapter(adapter);

                                //        for(int i=0; i < adapter.getGroupCount(); i++)
                                //            domainsList.expandGroup(i);

                                btnSave.setOnClickListener(new View.OnClickListener() {
                                    @Override
                                    public void onClick(View view) {
                                        HashMap<String, Integer> selectedDomains = new
HashMap<>();
                                        ArrayList<String> allDomains = new ArrayList<>();
                                        ArrayList<ArrayList<Integer>> allSubDomains = new
ArrayList<ArrayList<Integer>>();
```

```
                                selectedDomains = adapter.check_groups;
                                allDomains = (ArrayList<String>) adapter.groups;
                                allSubDomains = adapter.check_states;

                                HashMap<String, ArrayList<String>> updatedMap = new
HashMap<>();
                                String msg = "";
                                for(int i=0;i<allDomains.size();i++) {
                                   String currDomain = allDomains.get(i);
                                   ArrayList<String> temp2 = new ArrayList<>();
                                   if(selectedDomains.get(currDomain) == 1) {
                                      // domain is selected
                                      // Now, check for subdomains
                                      msg += currDomain;
                                      msg += " ";
                                      ArrayList<Integer> temp = allSubDomains.get(i);
                                      for(int j=0;j<temp.size();j++) {
                                         String currSubdomain =
adapter.children.get(i).get(j);

                                         if (temp.get(j) == 1) {
                                            // subdomain is also selected
                                            msg += currSubdomain;
                                            temp2.add(currSubdomain);
                                            msg += " ";
                                         }
                                      }
                                      updatedMap.put(currDomain, temp2);
                                   }
                                   msg += "\n";
                                }
//                                 Toast.makeText(MyDomain.this, msg,
Toast.LENGTH_SHORT).show();
                                Toast.makeText(MyDomain.this, "Details Saved!",
Toast.LENGTH_SHORT).show();
                                Log.d(TAG, "Lastly reg Id is "+RegID);

                                // now, update the document ib the database
                                DocumentReference docR_2 =
db.document("year/"+year+"- "+(year+1)+"/Users/"+RegID);
                                Map<String, Object> userObj = new HashMap<>();
                                userObj.put("Domains", null);
                                docR_2.set(userObj, SetOptions.merge());
                                userObj.put("Domains", updatedMap);
                                docR_2.set(userObj, SetOptions.merge());
                                Log.d(TAG, updatedMap.toString());

                                if (activity.equals("Register")) {
```

```
                                        startActivity(new Intent(getApplicationContext(),
joingroup.class));
                                    }

                                }
                            });

                        }
                        else {
                            Toast.makeText(MyDomain.this, "Could not find User details!",
Toast.LENGTH_SHORT).show();
                        }
                    }
                });
            }
        });
        }
        else {
            Log.d(TAG, "Error getting messages");
        }
    }
});
Log.d(TAG, groups.toString());
Log.d(TAG, "Last");
}

private interface FirestoretoreCallback {
    void callback(List<String> list1, ArrayList<ArrayList<String>> list2);

}

private void test() {

}

}
```

ProjectsFragment.java

```
package com.example.teproject;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.os.Handler;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
```

```
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.firebase.ui.firestore.FirestoreRecyclerOptions;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.Query;
import com.google.firebase.firestore.QuerySnapshot;
import com.google.gson.Gson;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Calendar;
import java.util.List;

public class ProjectsFragment extends Fragment {

    private static final String TAG = "ProjectsFragment";
    private View thisView;
    private int year;
```

```
    private RecyclerView recyclerView;

    private FirebaseFirestore db;
    private CollectionReference groupsRef;
    private ListGroupsAdapter adapter;
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {

        // Defines the xml file for the fragment
        thisView = inflater.inflate(R.layout.fragment_projects, container, false);

        Calendar calendar = Calendar.getInstance();
        year = calendar.get(Calendar.YEAR);
        db = FirebaseFirestore.getInstance();
        groupsRef = db.collection("year/"+year+"- "+(year+1)+"/Groups");
        recyclerView = thisView.findViewById(R.id.groups_recycler_view);
        recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));
        return thisView;
    }

    @Override
    public void onStart() {
        super.onStart();
        Query query = groupsRef;
        FirestoreRecyclerOptions<GroupsOverview> options = new
FirestoreRecyclerOptions.Builder<GroupsOverview>()
                .setQuery(query, GroupsOverview.class)
                .build();
        Log.d("Options: ", options.toString());
        adapter = new ListGroupsAdapter(options);
        recyclerView.setAdapter(adapter);
        adapter.setOnGroupItemClickListener(new
ListGroupsAdapter.OnGroupItemClickListener() {
            @Override
            public void onGrpItemClick(DocumentSnapshot documentSnapshot, int position) {
                GroupsOverview groupsOverview =
documentSnapshot.toObject(GroupsOverview.class);
//                Toast.makeText(getContext(), "Position: "+position+" ID:
"+documentSnapshot.getId(), Toast.LENGTH_SHORT).show();
                // start new activity
                Intent groupIntent = new Intent(getContext(), GroupProfile.class);
                groupIntent.putExtra("activity", "ProjectsFragment");

                // passing the clicked object's data...
                Gson gson = new Gson();
                String currUser = gson.toJson(groupsOverview);
```

```
                groupIntent.putExtra("groupsOverView", currUser);
                groupIntent.putExtra("groupId", documentSnapshot.getId());
                startActivity(groupIntent);
            }
        });
        adapter.startListening();
    }
}
```

UserOverview.java

```
package com.example.teproject;

import java.util.ArrayList;
import java.util.HashMap;

public class UserOverview {
    private String Branch;
    private HashMap<String, ArrayList<String>> Domains;
    private String Email;
    private String Github;
    private String GroupID;
    private String Linkedin;
    private String Name;
    private String Password;
    private String PhoneNo;
    private String RegistrationID;
    private String Resume;
    private boolean Role;
    private String RollNo;

    public UserOverview() {
    }

    public UserOverview(String branch, HashMap<String, ArrayList<String>> domains, String
email, String github, String groupID, String linkedin, String name, String password, String
phoneNo, String registrationID, String resume, boolean role, String rollNo) {
        Branch = branch;
        Domains = domains;
        Email = email;
        Github = github;
        GroupID = groupID;
        Linkedin = linkedin;
        Name = name;
        Password = password;
        PhoneNo = phoneNo;
        RegistrationID = registrationID;
        Resume = resume;
```

```
    Role = role;
    RollNo = rollNo;
}

public String getBranch() {
    return Branch;
}

public void setBranch(String branch) {
    Branch = branch;
}

public HashMap<String, ArrayList<String>> getDomains() {
    return Domains;
}

public void setDomains(HashMap<String, ArrayList<String>> domains) {
    Domains = domains;
}

public String getEmail() {
    return Email;
}

public void setEmail(String email) {
    Email = email;
}

public String getGithub() {
    return Github;
}

public void setGithub(String github) {
    Github = github;
}

public String getGroupID() {
    return GroupID;
}

public void setGroupID(String groupID) {
    GroupID = groupID;
}

public String getLinkedin() {
    return Linkedin;
}
```

```
public void setLinkedin(String linkedin) {
    Linkedin = linkedin;
}

public String getName() {
    return Name;
}

public void setName(String name) {
    Name = name;
}

public String getPassword() {
    return Password;
}

public void setPassword(String password) {
    Password = password;
}

public String getPhoneNo() {
    return PhoneNo;
}

public void setPhoneNo(String phoneNo) {
    PhoneNo = phoneNo;
}

public String getRegistrationID() {
    return RegistrationID;
}

public void setRegistrationID(String registrationID) {
    RegistrationID = registrationID;
}

public String getResume() {
    return Resume;
}

public void setResume(String resume) {
    Resume = resume;
}

public boolean getRole() {
    return Role;
}
```

```
    public void setRole(boolean role) {
        Role = role;
    }

    public String getRollNo() {
        return RollNo;
    }

    public void setRollNo(String rollNo) {
        RollNo = rollNo;
    }
}
```

editprofile.java

```
package com.example.teproject;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

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

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.SetOptions;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;

public class editprofile extends AppCompatActivity {

    private static final String CALLING_TAG = "MainActivity";
    TextView email, phone, linkedin, github, resume;
    Button savechanges;
```

```
    FirebaseFirestore fStore;
    FirebaseAuth fAuth;

    String RegID;
    private Toolbar mToolbar;

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

        mToolbar = findViewById(R.id.back_toolbar);
        mToolbar.setNavigationIcon(R.drawable.ic_arrow);
        mToolbar.setNavigationOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View view) {

                // Your code
                Intent profileIntent = new Intent(getApplicationContext(), myprofile.class);
                profileIntent.putExtra("caller", CALLING_TAG);
                startActivity(profileIntent);
                finish();
            }
        });

        Calendar calendar = Calendar.getInstance();
        int year = calendar.get(Calendar.YEAR);

        email = findViewById(R.id.edittext1);
        phone = findViewById(R.id.edittext2);
        linkedin = findViewById(R.id.edittext3);
        github = findViewById(R.id.edittext4);
        resume = findViewById(R.id.edittext5);

        savechanges = findViewById(R.id.savechanges);

        fAuth = FirebaseAuth.getInstance();
        fStore = FirebaseFirestore.getInstance();

        DocumentReference docR = fStore.document("year/"+year+"-
"+(year+1)+"/IDS/"+fAuth.getCurrentUser().getUid());

        docR.get().addOnSuccessListener(this, new
OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                if(documentSnapshot.exists()){
```

```
                RegID = documentSnapshot.getString("RegID");
                tp(year);
            }
        }
    });

    savechanges.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String email_temp = email.getText().toString().trim();
            String phone_temp = phone.getText().toString();
            String github_temp = github.getText().toString();
            String linkedin_temp = linkedin.getText().toString();
            String resume_temp = resume.getText().toString();
            Log.d("lalalala",email_temp);
            DocumentReference docR_2 = fStore.document("year/"+year+"-
"+(year+1)+"/Users/"+RegID);

            Map<String, Object> datatosave = new HashMap<>();
            datatosave.put("Email", email_temp);
            datatosave.put("PhoneNo", phone_temp);
            datatosave.put("Github", github_temp);
            datatosave.put("Linkedin", linkedin_temp);
            datatosave.put("Resume", resume_temp);

            docR_2.set(datatosave, SetOptions.merge()).addOnSuccessListener(new
OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void aVoid) {
                    Intent profileIntent = new Intent(getApplicationContext(), myprofile.class);
                    profileIntent.putExtra("caller", "MainActivity");
                    startActivity(profileIntent);
                }
            });




        }
    });
}

void tp(int year){
    DocumentReference docR_2 = fStore.document("year/"+year+"-
"+(year+1)+"/Users/"+RegID);
    docR_2.get().addOnSuccessListener(this, new
OnSuccessListener<DocumentSnapshot>() {
```

```
        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            if(documentSnapshot.exists()) {
                email.setText(documentSnapshot.getString("Email"));
                phone.setText(documentSnapshot.getString("PhoneNo"));
                linkedin.setText(documentSnapshot.getString("Linkedin"));
                github.setText(documentSnapshot.getString("Github"));
                resume.setText(documentSnapshot.getString("Resume"));
            }
        }
    });
  }

}
```

joingroup.java

```
package com.example.teproject;

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

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
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.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.SetOptions;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;
import java.util.Random;
```

```
import static com.google.firebase.firestore.FieldValue.arrayUnion;

public class joingroup extends AppCompatActivity {

    private TextView mCreateGroupText;
    private Button mCreateGrpBtn, mJoinGrpBtn, mSkipBtn;
    private Random rand;
    private EditText mJoinGroupTxt;
    private int year, codeint;
    final ArrayList<Integer> codeArr = new ArrayList<>();
    private String fireRegID, userUid, code, fireGrp;
    FirebaseFirestore fStore;
    FirebaseAuth fAuth;
    private DocumentReference docRef, docRef2;
    private boolean status, exist, fireRole, firegroupid;
    Toolbar mToolbar;

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

        mToolbar = findViewById(R.id.back_toolbar);
        mToolbar.setNavigationIcon(R.drawable.ic_arrow);
        mToolbar.setNavigationOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View view) {

                // Your code
                finish();
            }
        });

        Calendar calendar = Calendar.getInstance();
        year = calendar.get(Calendar.YEAR);

        mCreateGrpBtn = findViewById(R.id.creategroupbtn);
        mJoinGrpBtn = findViewById(R.id.joingroupbtn);
        mCreateGroupText = findViewById(R.id.groupCodetext);
        mJoinGroupTxt = findViewById(R.id.joingrptext);
        mSkipBtn = findViewById(R.id.skipbtn);
        rand = new Random();

        fStore = FirebaseFirestore.getInstance();
        fAuth = FirebaseAuth.getInstance();
        FirebaseUser user = fAuth.getCurrentUser();
        userUid =  user.getUid();
```

```
        String userEmail = user.getEmail();

        getIDS();
        docRef2 = fStore.document("year/"+year+"- "+(year+1)+"/Users/"+fireRegID);


        mCreateGrpBtn. setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                codeint = rand.nextInt((9999 - 100) + 1) + 10;
                code = Integer.toString(codeint);
                docRef = fStore.document("year/"+year+"- "+(year+1)+"/Groups/"+code);
                docRef.get().addOnCompleteListener(new
OnCompleteListener<DocumentSnapshot>() {
                    @Override
                    public void onComplete(@NonNull Task<DocumentSnapshot> task) {
                        DocumentSnapshot documentSnapshot = task.getResult();
                        while(documentSnapshot.exists()){
                            codeint = rand.nextInt((9999 - 100) + 1) + 10;
                            code = Integer.toString(codeint);
                            Log.d("TAG", "Code: "+ code);
                        }
                        if(!documentSnapshot.exists()){
                            mCreateGrpBtn.setEnabled(false);
                            mJoinGrpBtn.setEnabled(false);
                            Log.d("TAG", "Reg is: "+ fireRegID);

                            mCreateGroupText.setVisibility(View.VISIBLE);
                            mCreateGroupText.setText("Group code: "+code + "\n" +
                                    "Share it with group members only!!");
                            Log.d("TAG", "Code "+ code);
                            addGroupID();
                        }
                    }
                });
            }
        });

        mSkipBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getApplicationContext(), MainActivity.class));
            }
        });

        mJoinGrpBtn.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
        public void onClick(View v) {
            code = mJoinGroupTxt.getText().toString().trim();
            if(TextUtils.isEmpty(code)){
                mJoinGroupTxt.setError("Enter code");
            }
            docRef = fStore.document("year/"+year+"- "+(year+1)+"/Groups/"+code);
            docRef2 = fStore.document("year/"+year+"- "+(year+1)+"/Users/"+fireRegID);
            Log.d("TAG", "code is: "+ code);
            checkIfgroupExists();
        }
    });
}


void addMemberToGrp(){

    Map<String, Object> datatosave2 = new HashMap<>();
    if(fireRole){
        datatosave2.put("Members", arrayUnion(fireRegID));
    } else {
        datatosave2.put("MentorID", fireRegID);
    }

    docRef.set(datatosave2, SetOptions.merge()).addOnCompleteListener(new
OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            if(task.isSuccessful()){
                Log.d("TAG", "Member successfully added");
                Toast.makeText(joingroup.this, "You're a member of team: "+code,
Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getApplicationContext(), MainActivity.class));
                mSkipBtn.setText("Next");
            } else{
                Log.d("TAG", "Member not added!");
                mJoinGrpBtn.setEnabled(false);
                Toast.makeText(joingroup.this, "You're in another team",
Toast.LENGTH_SHORT).show();
            }
        }
    });
}

void addGroupID(){

    docRef2 = fStore.document("year/"+year+"- "+(year+1)+"/Users/"+fireRegID);
    docRef2.get().addOnCompleteListener(new
OnCompleteListener<DocumentSnapshot>() {
```

```
        @Override
        public void onComplete(@NonNull Task<DocumentSnapshot> task) {
            DocumentSnapshot document = task.getResult();
            if(document.exists()){
                Log.d("TAG", "Exissttsssss");
                String groupid = document.getString("GroupID");
                if(groupid.equals("N.A.")){
                    status = true;
                    Map<String, Object> datatosave = new HashMap<>();
                    datatosave.put("GroupID", code);
                    Log.d("TAG","groupid: "+ code);

                    docRef2.update(datatosave).addOnCompleteListener(new
OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if(task.isSuccessful()){
                                Log.d("Hooray", "GroupID added successfully");
                                addMemberToGrp();
                            }else{
                                Log.d("Error", "There was an error!");
                            }
                        }
                    });

                } else{
                    status = false;
                    mJoinGrpBtn.setEnabled(false);
                    Log.d("TAG", "Already a group member!");
                    Toast.makeText(joingroup.this, "Already a member",
Toast.LENGTH_SHORT).show();
                }
            }
        }
    });

  }
  void checkIfgroupExists(){
      docRef.get().addOnCompleteListener(new
OnCompleteListener<DocumentSnapshot>() {
        @Override
        public void onComplete(@NonNull Task<DocumentSnapshot> task) {
            DocumentSnapshot document = task.getResult();
            if(document.exists()){
                exist = true;
                mCreateGrpBtn.setEnabled(false);
                    addGroupID();
            } else{
```

```
                exist = false;
                mJoinGroupTxt.setError("Invalid code");
                mCreateGrpBtn.setEnabled(true);
            }
        }
    });
}

void getRegID(){
    DocumentReference dRef = fStore.document("year/"+year+"-
"+(year+1)+"/Users/"+fireRegID);

    dRef.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
        @Override
        public void onComplete(@NonNull Task<DocumentSnapshot> task) {
            DocumentSnapshot document = task.getResult();
            if (document.exists()) {
                fireRegID = document.getString("RegistrationID");
                fireRole = document.getBoolean("Role");
                fireGrp = document.getString("GroupID");
                if(fireRole){
                    mCreateGrpBtn.setEnabled(true);
                } else {
                    mCreateGrpBtn.setEnabled(false);
                }
                if(fireGrp.equals("N.A.")){
                    mCreateGrpBtn.setEnabled(true);
                    mJoinGrpBtn.setEnabled(true);
                } else {
                    mCreateGrpBtn.setEnabled(false);
                    mJoinGrpBtn.setEnabled(false);
                }

                Log.d("TAG", "data => " + document.getData());
                Log.d("TAG", "ROle is: "+ fireRole);
            } else {
                Log.d("TAG", "Error getting documents: ", task.getException());
            }

        }
    });
}
void getIDS(){
    DocumentReference dRef_1 = fStore.document("year/"+year+"-
"+(year+1)+"/IDS/"+userUid);
    dRef_1.get().addOnCompleteListener(new
OnCompleteListener<DocumentSnapshot>() {
        @Override
```

```
            public void onComplete(@NonNull Task<DocumentSnapshot> task) {
                DocumentSnapshot document = task.getResult();
                if(document.exists()){
                    fireRegID = document.getString("RegID");
                    Log.d("TAG", "RegID: "+ fireRegID);
                    getRegID();

                } else{
                    Log.d("TAG", "Error fetching the document");
                }
            }
        });
    }
}
```

myprofile.java

```
package com.example.teproject;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
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 com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.storage.FirebaseStorage;
```

```
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.google.gson.Gson;
import com.squareup.picasso.Picasso;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;

public class myprofile extends AppCompatActivity {
    private static final String TAG = "myprofile";

    StorageReference mStorageReference;
    FirebaseFirestore fStore;
    FirebaseAuth fAuth;
    String RegID;
    TextView fullname, emailid, phone, branch, rollno, role, groupid, linkedin, github, resume,
upload, regid;
    Button editProfBtn;
    TextView domains;
    ImageView profile_pic;
    private Toolbar mToolbar;

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

        mToolbar = findViewById(R.id.back_toolbar);
        mToolbar.setNavigationIcon(R.drawable.ic_arrow);
        mToolbar.setNavigationOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View view) {

                // Your code
                finish();
            }
        });

        Calendar calendar = Calendar.getInstance();
        int year = calendar.get(Calendar.YEAR);
        fullname = findViewById(R.id.fullname);
        emailid = findViewById(R.id.emailid);
        phone = findViewById(R.id.phone);
        branch = findViewById(R.id.branch);
        rollno = findViewById(R.id.rollno);
```

```
        role = findViewById(R.id.role);
        groupid = findViewById(R.id.groupid);
        linkedin = findViewById(R.id.linkedin);
        github = findViewById(R.id.github);
        resume = findViewById(R.id.resume);
        domains = findViewById(R.id.domaintext);
        regid = findViewById(R.id.RegID);

        upload = findViewById(R.id.upload);
        profile_pic = findViewById(R.id.profilepicture);

        editProfBtn = findViewById(R.id.editprofile);

        fAuth = FirebaseAuth.getInstance();
        fStore = FirebaseFirestore.getInstance();
        mStorageReference = FirebaseStorage.getInstance().getReference();

        // checking who has called this activity
        Intent intent = getIntent();
        String caller = intent.getStringExtra("caller");
        if (caller.equals("MainActivity")) {
            Log.d(TAG, "Main activity called me!");
            DocumentReference docR = fStore.document("year/"+year+"-
"+(year+1)+"/IDS/"+fAuth.getCurrentUser().getUid());

            docR.get().addOnSuccessListener(this, new
OnSuccessListener<DocumentSnapshot>() {
                @Override
                public void onSuccess(DocumentSnapshot documentSnapshot) {
                    if(documentSnapshot.exists()){
                        RegID = documentSnapshot.getString("RegID");
                        tp(year);
                    }
                }
            });

            editProfBtn.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    startActivity(new Intent(getApplicationContext(), editprofile.class));
                    finish();
                }
            });

            StorageReference fileref =
mStorageReference.child("users/"+fAuth.getCurrentUser().getUid()+"/profile.jpg");
            fileref.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                @Override
```

```
            public void onSuccess(Uri uri) {
                Picasso.get().load(uri).into(profile_pic);
            }
        });

        upload.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent openGallery = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                startActivityForResult(openGallery,1000);
            }
        });

    } else if(caller.equals("ListUserFragment")) {
        Log.d(TAG, "Fragment called me");
        editProfBtn.setVisibility(View.GONE);
        upload.setVisibility(View.GONE);

        // extracting the gson object
        Gson gson = new Gson();
        UserOverview userOverview =
gson.fromJson(getIntent().getStringExtra("userOverview"), UserOverview.class);
        Log.d(TAG, userOverview.getName());

        fullname.setText(userOverview.getName());
        emailid.setText(userOverview.getEmail());
        branch.setText(userOverview.getBranch());
        rollno.setText(userOverview.getRollNo());
        if(userOverview.getRole() == true){
            role.setText("Student");
        }
        else{
            role.setText("Teacher");
        }
        groupid.setText(userOverview.getGroupID());
        phone.setText(userOverview.getPhoneNo());
        linkedin.setText(userOverview.getLinkedin());
        github.setText(userOverview.getGithub());
        resume.setText(userOverview.getResume());
        regid.setText(userOverview.getRegistrationID());

        Map<String , ArrayList<String>> subDomains = new HashMap<>();
        subDomains = userOverview.getDomains();
        String s = "";
        if(subDomains==null){
        }
        else {
```

```
            int ct = 0;
            for (Map.Entry m: subDomains.entrySet()){
                if(ct!=0){
                    s += "\n\n";
                }
                ct++;
                String dname = m.getKey().toString();
                ArrayList<String> snames = (ArrayList<String>) m.getValue();
                s += dname + " : ";
                int i=0;
                for(i=0; i< snames.size(); i++ ){
                    s += snames.get(i)+"   ";
                }
            }
            Log.d("hii", s);
            domains.setText(s);
        }

    }

}
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode == 1000){
        if(resultCode == Activity.RESULT_OK){
            Uri imageuri = data.getData();
            profile_pic.setImageURI(imageuri);

            uploadimagetofirebase(imageuri);
        }
    }
}

private void uploadimagetofirebase(Uri imageuri) {
    final StorageReference fileref =
mStorageReference.child("users/"+fAuth.getCurrentUser().getUid()+"/profile.jpg");
    fileref.putFile(imageuri).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
            Toast.makeText(myprofile.this,"Image
uploaded",Toast.LENGTH_SHORT).show();
            fileref.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                @Override
                public void onSuccess(Uri uri) {
                    Picasso.get().load(uri).into(profile_pic);
                }
```

```
            });
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText(myprofile.this,"Image could not be
uploaded",Toast.LENGTH_SHORT).show();
        }
    });
}
void tp(int year){
    DocumentReference docR_2 = fStore.document("year/"+year+"-
"+(year+1)+"/Users/"+RegID);
    docR_2.get().addOnSuccessListener(this, new
OnSuccessListener<DocumentSnapshot>() {

        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            if(documentSnapshot.exists()) {
                fullname.setText(documentSnapshot.getString("Name"));
                emailid.setText(documentSnapshot.getString("Email"));
                branch.setText(documentSnapshot.getString("Branch"));
                rollno.setText(documentSnapshot.getString("RollNo"));
                if(documentSnapshot.getBoolean("Role") == true){
                    role.setText("Student");
                }
                else{
                    role.setText("Teacher");
                }
                groupid.setText(documentSnapshot.getString("GroupID"));
                phone.setText(documentSnapshot.getString("PhoneNo"));
                linkedin.setText(documentSnapshot.getString("Linkedin"));
                github.setText(documentSnapshot.getString("Github"));
                resume.setText(documentSnapshot.getString("Resume"));
                regid.setText(documentSnapshot.getString("RegistrationID"));

                Map<String , ArrayList<String>> subDomains = new HashMap<>();
                subDomains = (Map<String , ArrayList<String>>)
documentSnapshot.get("Domains");
                String s = "";
                if(subDomains==null){
                }
                else {
                    int ct=0;
                    for (Map.Entry m: subDomains.entrySet()){
                        if(ct!=0){
                            s += "\n\n";
                        }
```

```
                    ct++;
                    String dname = m.getKey().toString();
                    ArrayList<String> snames = (ArrayList<String>) m.getValue();
                    s += dname + " : ";
                    for(int i=0; i< snames.size(); i++ ){
                        s += snames.get(i)+"   ";
                    }
                }
                Log.d("hii", s);
                domains.setText(s);
            }
        }
    }
});
  }
}
```