MainActivity.Java

```java
package com.example.spllabportal;
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 androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.auth.api.signin.GoogleSignIn;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInClient;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthCredential;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.GoogleAuthProvider;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.HashMap;

public class MainActivity extends AppCompatActivity {

   private static final String TAG = "GoogleSignIn";
   private static final int RC_SIGN_IN = 9001;
   private GoogleSignInClient mGoogleSignInClient;
   private FirebaseAuth mAuth;
   private Button signInButton;
   private TextView welcomeTextView;
   private TextView nameTextView;

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

       signInButton = findViewById(R.id.signinbutton);
       welcomeTextView = findViewById(R.id.welcome);
       nameTextView = findViewById(R.id.name);
       mAuth = FirebaseAuth.getInstance();
```

```
        // Configure Google Sign-In
        GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)

.requestIdToken("1006634308462-ecn0hegen139lpm0vfdlmhett0cced0m.apps.googleuse
rcontent.com")
                .requestEmail()
                .build();

        mGoogleSignInClient = GoogleSignIn.getClient(this, gso);

        signInButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                signOutAndSignIn(); // Sign out and then sign in again to show
account chooser
            }
        });
    }

    private void signOutAndSignIn() {
        // Sign out before showing the account chooser
        mGoogleSignInClient.signOut()
                .addOnCompleteListener(this, task -> {
                    // After signing out, trigger the sign-in flow again
                    signIn();
                });
    }

    private void signIn() {
        Intent signInIntent = mGoogleSignInClient.getSignInIntent();
        startActivityForResult(signInIntent, RC_SIGN_IN);
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data)
{
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == RC_SIGN_IN) {
            Task<GoogleSignInAccount> task =
GoogleSignIn.getSignedInAccountFromIntent(data);
            handleSignInResult(task);
        }
    }

    private void handleSignInResult(Task<GoogleSignInAccount> completedTask) {
        try {
```

```
            GoogleSignInAccount account =
completedTask.getResult(ApiException.class);
            if (account != null) {
                String email = account.getEmail();
                if (email != null && email.endsWith("@bitsathy.ac.in")) {
                    AuthCredential credential =
GoogleAuthProvider.getCredential(account.getIdToken(), null);

                    mAuth.signInWithCredential(credential)
                            .addOnCompleteListener(this, task -> {
                                if (task.isSuccessful()) {
                                    FirebaseUser user = mAuth.getCurrentUser();
                                    if (user != null) {
                                        String userEmail = user.getEmail();
                                        boolean isStudent =
isStudentEmail(userEmail);
                                        String userName =
user.getDisplayName(); // ✅ Get Name instead of extracting ID

                                        Log.d(TAG, "Retrieved Name: " +
userName);

                                        DatabaseReference labRef =
FirebaseDatabase.getInstance().getReference("SpecialLab");

labRef.get().addOnCompleteListener(labTask -> {
                                            if (labTask.isSuccessful() &&
labTask.getResult().exists()) {
                                                boolean userExists = false;


                                                for (DataSnapshot lab :
labTask.getResult().getChildren()) {
                                                    for (DataSnapshot userEntry
: lab.getChildren()) {
                                                        String storedName =
userEntry.child("Name").getValue(String.class);
                                                        Log.d(TAG, "Checking: "
+ storedName);

                                                        if (storedName != null
&& storedName.equalsIgnoreCase(userName)) {
                                                            Log.d(TAG, "User
Found in: " + lab.getKey());
                                                            userExists = true;
                                                            break;
                                                        }
                                                    }
                                                    if (userExists) break; //
Exit loop early if user is found
```

```
                                                }

                                                if (userExists) {
                                                    Log.d(TAG, "User exists.
Redirecting...");
                                                    if (isStudent) {
                                                        startActivity(new
Intent(MainActivity.this, UserModeActivity.class));
                                                    } else {
                                                        startActivity(new
Intent(MainActivity.this, AdminModeActivity.class));
                                                    }
                                                } else {
                                                    Log.d(TAG, "User not found.
Redirecting to RegistrationActivity.");
                                                    startActivity(new
Intent(MainActivity.this, RegistrationActivity.class));
                                                }
                                            } else {
                                                Log.d(TAG, "No SpecialLab
found. Redirecting to Registration.");
                                                startActivity(new
Intent(MainActivity.this, RegistrationActivity.class));
                                            }
                                            finish();
                                        });
                                    }
                                } else {
                                    Toast.makeText(this, "Firebase
Authentication failed.", Toast.LENGTH_SHORT).show();
                                }
                            });
                } else {
                    Toast.makeText(this, "Invalid email domain.",
Toast.LENGTH_SHORT).show();
                }
            }
        } catch (ApiException e) {
            Log.w(TAG, "signInResult:failed code=" + e.getStatusCode());
            Toast.makeText(this, "Sign-in failed. Please try again.",
Toast.LENGTH_SHORT).show();
        }
    }

    private boolean isStudentEmail(String email) {
        return
email.matches("^[a-zA-Z]+\\.[a-zA-Z]{2}\\d{2}@bitsathy\\.ac\\.in$");
    }
```

```
    private boolean isFacultyEmail(String email) {
        return email.matches("^[a-zA-Z]+@bitsathy\\.ac\\.in$");
    }
}
```

Project_entry —> fragment

```
package com.example.spllabportal;




import android.content.Context;
import android.graphics.Typeface;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.Spinner;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AlertDialog;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.RecyclerView;

import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.textfield.TextInputEditText;
import com.google.android.material.textfield.TextInputLayout;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
```

```
import java.util.HashMap;
import java.util.List;
import java.util.Map;
public class ProjectEntryFragment extends Fragment {
   private DatabaseReference databaseReference;
   private FloatingActionButton fabAddProject;
   private FirebaseAuth mAuth;
   private String loggedInUserName;
   private TableLayout projectTable;
   private Spinner spinnerGuideName; // Guide Name Spinner
   private DatabaseReference specialLabReference; // Firebase reference for
faculty under Special Labs
   private List<String> facultyList; // List to store faculty names
   private ArrayAdapter<String> facultyAdapter; // Adapter for the spinner

   public ProjectEntryFragment() {
       // Required empty public constructor
   }

   @Nullable
   @Override
   public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
       View view = inflater.inflate(R.layout.fragment_project_entry,
container, false);
       specialLabReference =
FirebaseDatabase.getInstance().getReference("SpecialLab");
       facultyList = new ArrayList<>();
       facultyAdapter = new ArrayAdapter<>(requireContext(),
android.R.layout.simple_spinner_item, facultyList);

facultyAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdow
n_item);

       mAuth = FirebaseAuth.getInstance();
       FirebaseUser user = mAuth.getCurrentUser();
       databaseReference =
FirebaseDatabase.getInstance().getReference("Projects");
       if (user != null) {
           loggedInUserName = user.getDisplayName();
       } else {
           loggedInUserName = "";
       }
       fabAddProject = view.findViewById(R.id.fabAddProject);
       fabAddProject.setOnClickListener(v -> showAddProjectDialog());
       projectTable = view.findViewById(R.id.projectTable);
       loadProjects();
       return view;
   }
```

```
   private void loadProjects() {
       databaseReference.addValueEventListener(new ValueEventListener() {
           @Override
           public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
               if (getContext() == null) {
                   return;  // Prevents crash if fragment is not attached
               }
               projectTable.removeAllViews();
               TableRow headerRow = new TableRow(getContext());
               addTextViewToRow(headerRow, "Team Leader", true);
               addTextViewToRow(headerRow, "Team Leader Roll Number", true);
               addTextViewToRow(headerRow, "Team Members Name", true);
               addTextViewToRow(headerRow, "Roll Numbers", true);
               addTextViewToRow(headerRow, "Mail IDs", true);
               addTextViewToRow(headerRow, "Department", true);
               addTextViewToRow(headerRow, "Year", true);
               addTextViewToRow(headerRow, "Special Lab", true);
               addTextViewToRow(headerRow, "Guide Name", true);
               addTextViewToRow(headerRow, "Duration", true);
               addTextViewToRow(headerRow, "Project Title", true);
               addTextViewToRow(headerRow, "Project Description", true);
               projectTable.addView(headerRow);
               for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
                   Map<String, Object> projectData = (Map<String, Object>)
snapshot.getValue();
                   String teamLeaderName = (String)
projectData.get("teamLeaderName");
                   if (teamLeaderName != null &&
teamLeaderName.equalsIgnoreCase(loggedInUserName)) {
                       String Roll = (String) projectData.get("rollNumber");
                       String Department = (String)
projectData.get("department");
                       String Year = (String) projectData.get("year");
                       String specialLab = (String)
projectData.get("specialLab");
                       String GuideName = (String)
projectData.get("guideName");
                       String Duration = (String) projectData.get("Duration");
                       String projectTitle = (String)
projectData.get("projectTitle");
                       String projectDescription = (String)
projectData.get("projectDescription");
                       ArrayList<String> teamMemberNames = new ArrayList<>();
                       ArrayList<String> teamMemberRollNumbers = new
ArrayList<>();
                       ArrayList<String> teamMemberMailid = new ArrayList<>();
                       DataSnapshot teamMembersSnapshot =
snapshot.child("teamMembers");
```

```
                    for (DataSnapshot memberSnapshot :
teamMembersSnapshot.getChildren()) {
                        String memberName =
memberSnapshot.child("name").getValue(String.class);
                        String memberRollNumber =
memberSnapshot.child("rollNumber").getValue(String.class);
                        String memberMailId =
memberSnapshot.child("mailId").getValue(String.class);

                        if (memberName != null) {
                            teamMemberNames.add(memberName);
                        }
                        if (memberRollNumber != null) {
                            teamMemberRollNumbers.add(memberRollNumber);
                        }
                        if (memberMailId != null) {
                            teamMemberMailid.add(memberMailId);
                        }
                    }

                    TableRow dataRow = new TableRow(getContext());
                    // Team Leader & Team Members Details
                    addTextViewToRow(dataRow, teamLeaderName, false);
                    addTextViewToRow(dataRow, Roll, false);
                    addTextViewToRow(dataRow, String.join(", ",
teamMemberNames), false);
                    addTextViewToRow(dataRow, String.join(", ",
teamMemberRollNumbers), false);
                    addTextViewToRow(dataRow, String.join(", ",
teamMemberMailid), false);
                    addTextViewToRow(dataRow, Department, false);
                    addTextViewToRow(dataRow, Year, false);
                    addTextViewToRow(dataRow, specialLab, false);
                    addTextViewToRow(dataRow, GuideName, false);
                    addTextViewToRow(dataRow, Duration, false);
                    addShowContentTextView(dataRow, "Show Content →",
projectTitle);
                    addShowContentTextView(dataRow, "Show Content →",
projectDescription);


                    projectTable.addView(dataRow);
                }
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {
```

```
                Toast.makeText(getContext(), "Error loading projects: " +
databaseError.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void addTextViewToRow(TableRow row, String text, boolean isHeader)
{
        TextView textView = new TextView(getContext());
        textView.setLayoutParams(new TableRow.LayoutParams(
                TableRow.LayoutParams.WRAP_CONTENT,
                TableRow.LayoutParams.WRAP_CONTENT,
                1f  // Adjust weight to distribute space evenly
        ));
        textView.setText(text);
        textView.setPadding(12, 8, 12, 8);
        int backgroundColor;
        int textColor;
        if (isHeader) {
            backgroundColor = ContextCompat.getColor(getContext(),
R.color.teal_700); // Teal background for headers
            textColor = ContextCompat.getColor(getContext(), R.color.white);
            textView.setTypeface(null, Typeface.BOLD);
        } else {
            backgroundColor = ContextCompat.getColor(getContext(),
R.color.white); // White background for data rows
            textColor = ContextCompat.getColor(getContext(), R.color.black);
        }
        textView.setBackgroundColor(backgroundColor);
        textView.setTextColor(textColor);
        row.addView(textView);
    }

    private void addShowContentTextView(TableRow row, String label, String
content) {
        TextView textView = new TextView(getContext());
        textView.setText(label);
        int backgroundColor;
        backgroundColor = ContextCompat.getColor(getContext(), R.color.white);
        textView.setBackgroundColor(backgroundColor);
        textView.setTextColor(getResources().getColor(android.R.color.black));
        textView.setPadding(12, 8, 12, 8);
        textView.setOnClickListener(v -> showContentDialog(content));
        row.addView(textView);
    }

    private void showContentDialog(String content) {
        AlertDialog.Builder builder = new
AlertDialog.Builder(requireContext());
```

```
        builder.setTitle("Content Details");
        builder.setMessage(content);
        builder.setPositiveButton("Close", (dialog, which) ->
dialog.dismiss());
        builder.show();
    }

    private void loadFacultyNames() {
        DatabaseReference facultyRef =
FirebaseDatabase.getInstance().getReference("SpecialLab").child("Faculty");

        facultyRef.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                facultyList.clear(); // Clear previous data

                for (DataSnapshot facultySnapshot : snapshot.getChildren()) {
                    String facultyName =
facultySnapshot.child("Name").getValue(String.class);
                    if (facultyName != null) {
                        facultyList.add(facultyName);
                    }
                }

                // Notify adapter that data has changed
                facultyAdapter.notifyDataSetChanged();
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(requireContext(), "Failed to load faculty
names", Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void showAddProjectDialog() {
        AlertDialog.Builder builder = new
AlertDialog.Builder(requireContext());
        builder.setTitle("Register New Project");

        View view =
LayoutInflater.from(getContext()).inflate(R.layout.dialog_add_project, null);
        builder.setView(view);

        Spinner spinnerDuration = view.findViewById(R.id.spinnerDuration);
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
                requireContext(),
                R.array.Durations,
```

```
                android.R.layout.simple_spinner_item
        );

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
;
        spinnerDuration.setAdapter(adapter);

        spinnerGuideName = view.findViewById(R.id.spinnerGuide);

        // Ensure facultyList is initialized
        facultyList = new ArrayList<>();
        facultyAdapter = new ArrayAdapter<>(requireContext(),
android.R.layout.simple_spinner_item, facultyList);

facultyAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdow
n_item);

        // Set adapter initially (empty list)
        spinnerGuideName.setAdapter(facultyAdapter);
        EditText etTeamLeaderName = view.findViewById(R.id.etTeamLeaderName);
        EditText etTeamMemberCount = view.findViewById(R.id.etTeamMemberCount);
        EditText etRollNumber = view.findViewById(R.id.etRollNumber);
        EditText etSpecialLab = view.findViewById(R.id.etSpecialLab);
        EditText etDepartment = view.findViewById(R.id.etDepartment);
        EditText etYear = view.findViewById(R.id.etYear);
        LinearLayout teamMembersContainer =
view.findViewById(R.id.teamMembersContainer);
        EditText etProjectTitle = view.findViewById(R.id.etProjectTitle);
        EditText etProjectDescription =
view.findViewById(R.id.etProjectDescription);
        etTeamLeaderName.setText(loggedInUserName);
        etTeamLeaderName.setEnabled(false);

        loadFacultyNames();
        ArrayList<TextInputEditText> teamMemberInputs = new ArrayList<>();

        // Dynamic Team Member Field Generation
        etTeamMemberCount.setOnFocusChangeListener((v, hasFocus) -> {
            if (!hasFocus) {
                teamMembersContainer.removeAllViews();
                String countStr =
etTeamMemberCount.getText().toString().trim();
                if (!countStr.isEmpty()) {
                    int count = Integer.parseInt(countStr);
                    for (int i = 0; i < count; i++) {
                        LinearLayout memberLayout = new
LinearLayout(getContext());
                        memberLayout.setOrientation(LinearLayout.VERTICAL);
```

```
                        TextInputLayout nameLayout = new
TextInputLayout(getContext());
                        TextInputEditText nameInput = new
TextInputEditText(getContext());
                        nameInput.setHint("Team Member " + (i + 1));
                        nameLayout.addView(nameInput);
                        TextInputLayout rollLayout = new
TextInputLayout(getContext());
                        TextInputEditText rollInput = new
TextInputEditText(getContext());
                        rollInput.setHint("Roll Number " + (i + 1));
                        rollLayout.addView(rollInput);
                        TextInputLayout labLayout = new
TextInputLayout(getContext());
                        TextInputEditText labInput = new
TextInputEditText(getContext());
                        labInput.setHint("Mail id of " + (i + 1));
                        labLayout.addView(labInput);
                        memberLayout.addView(nameLayout);
                        memberLayout.addView(rollLayout);
                        memberLayout.addView(labLayout);
                        teamMembersContainer.addView(memberLayout);

                        teamMemberInputs.add(nameInput);
                        teamMemberInputs.add(rollInput);
                        teamMemberInputs.add(labInput);
                    }
                }
            }
        });
        builder.setPositiveButton("Submit", (dialog, which) -> {
            String teamLeaderName =
etTeamLeaderName.getText().toString().trim();
            String rollNumber = etRollNumber.getText().toString().trim();
            String specialLab = etSpecialLab.getText().toString().trim();
            String department = etDepartment.getText().toString().trim();
            String year = etYear.getText().toString().trim();
            String selectedDuration =
spinnerDuration.getSelectedItem().toString();
            String guideName = spinnerGuideName.getSelectedItem().toString();
            String projectTitle = etProjectTitle.getText().toString().trim();
            String projectDescription =
etProjectDescription.getText().toString().trim();
            ArrayList<String> teamMembers = new ArrayList<>();
            for (TextInputEditText input : teamMemberInputs) {
                String memberInfo = input.getText().toString().trim();
                if (!TextUtils.isEmpty(memberInfo)) {
                    teamMembers.add(memberInfo);
                }
```

```
            }
            if (TextUtils.isEmpty(teamLeaderName) ||
TextUtils.isEmpty(rollNumber) || TextUtils.isEmpty(projectTitle)) {
                Toast.makeText(getContext(), "Team Leader Name, Roll Number,
and Project Title are required!", Toast.LENGTH_SHORT).show();
                return;
            }
            // Check if the logged-in user is the Team Leader
            if (!teamLeaderName.equalsIgnoreCase(loggedInUserName)) {
                Toast.makeText(getContext(), "Only the Team Leader can submit
this project!", Toast.LENGTH_SHORT).show();
                return;
            }// Save Project Data to Firebase
            saveProjectToFirebase(rollNumber, teamLeaderName, teamMembers,
specialLab, guideName, selectedDuration, department, year, projectTitle,
projectDescription);
        }).setNegativeButton("Cancel", (dialog, which) -> dialog.dismiss());
        builder.show();
    }

    private void saveProjectToFirebase(String rollNumber, String
teamLeaderName, ArrayList<String> teamMembers, String specialLab, String
guideName, String duration,
                                       String department, String year, String
projectTitle,
                                       String projectDescription) {

        String projectId = databaseReference.push().getKey(); // Unique Project
ID
        Map<String, Object> projectData = new HashMap<>();
        projectData.put("teamLeaderName", loggedInUserName);
        projectData.put("rollNumber", rollNumber);
        projectData.put("department", department);
        projectData.put("specialLab", specialLab);
        projectData.put("Duration", duration);
        projectData.put("guideName", guideName);
        projectData.put("year", year);
        projectData.put("projectTitle", projectTitle);
        projectData.put("projectDescription", projectDescription);
        Map<String, Object> teamMembersMap = new HashMap<>();
        int memberIndex = 1;
        for (int i = 0; i < teamMembers.size(); i += 3) { // Each member has 3
values: name, rollNumber, mailId
            Map<String, String> memberDetails = new HashMap<>();
            memberDetails.put("name", teamMembers.get(i));
            memberDetails.put("rollNumber", teamMembers.get(i + 1));
            memberDetails.put("mailId", teamMembers.get(i + 2));

            teamMembersMap.put("member" + memberIndex, memberDetails);
```

```
            memberIndex++;
        }

        projectData.put("teamMembers", teamMembersMap);
        databaseReference.child(rollNumber).setValue(projectData)
                .addOnSuccessListener(aVoid -> Toast.makeText(getContext(),
"Project Registered Successfully!", Toast.LENGTH_SHORT).show())
                .addOnFailureListener(e -> Toast.makeText(getContext(), "Failed
to Register Project: " + e.getMessage(), Toast.LENGTH_SHORT).show());
    }

}
```

TaskManagement —> fragment

```
package com.example.spllabportal;

import android.app.DatePickerDialog;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

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

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.Calendar;

public class TaskManagementFragment extends Fragment {

    private EditText etStartDate;
    private ImageView ivCalendarIcon;
```

```
    private FirebaseAuth mAuth;
    private DatabaseReference mDatabase;

    private String loggedInUserName; // Holds the logged-in user's name

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

        etStartDate = view.findViewById(R.id.etStartDate);
        ivCalendarIcon = view.findViewById(R.id.ivCalendarIcon);

        // Initialize Firebase Auth and Database
        mAuth = FirebaseAuth.getInstance();
        mDatabase = FirebaseDatabase.getInstance().getReference("Projects");

        // Retrieve the logged-in user's name
        FirebaseUser user = mAuth.getCurrentUser();
        if (user != null) {
            loggedInUserName = user.getDisplayName(); // Get the user's display
name
        }

        // Set up DatePickerDialog on calendar icon click
        ivCalendarIcon.setOnClickListener(v -> showDatePickerDialog());

        return view;
    }

    private void showDatePickerDialog() {
        // Get current date
        Calendar calendar = Calendar.getInstance();
        int year = calendar.get(Calendar.YEAR);
        int month = calendar.get(Calendar.MONTH);
        int day = calendar.get(Calendar.DAY_OF_MONTH);

        // Show DatePickerDialog
        DatePickerDialog datePickerDialog = new DatePickerDialog(getContext(),
                (view, year1, month1, dayOfMonth) -> {
                    // Format the selected date
                    String selectedDate = dayOfMonth+ "-"+ (month1 +
1)+"-"+year1;
                    etStartDate.setText(selectedDate);  // Show selected date
in EditText
                    storeStartDate(selectedDate);  // Save selected date to
Firebase
```

```
            }, year, month, day);
        datePickerDialog.show();
    }

    private void storeStartDate(final String selectedDate) {
        if (loggedInUserName != null) {
            // Read the entire "Projects" node
            mDatabase.addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {
                    boolean projectFound = false;

                    // Loop through all child nodes inside "Projects"
                    for (DataSnapshot projectSnapshot :
dataSnapshot.getChildren()) {
                        // Get the TeamLeaderName for the current child node
                        String teamLeader =
projectSnapshot.child("teamLeaderName").getValue(String.class);
                        String guideName =
projectSnapshot.child("guideName").getValue(String.class); // Fetch Guide Name
                        String duration =
projectSnapshot.child("Duration").getValue(String.class); // Fetch Duration

                        // If the logged-in user is the team leader, update
StartDate
                        if (teamLeader != null &&
teamLeader.equals(loggedInUserName)) {

projectSnapshot.getRef().child("StartDate").setValue(selectedDate)
                                    .addOnSuccessListener(aVoid -> {
                                        Toast.makeText(getContext(), "Start
Date updated successfully", Toast.LENGTH_SHORT).show();
                                    })
                                    .addOnFailureListener(e -> {
                                        Toast.makeText(getContext(), "Failed to
update Start Date", Toast.LENGTH_SHORT).show();
                                    });



                            projectFound = true;
                            break; // Stop after finding the matching project
                        }
                    }

                    // If no project was found for the logged-in user
                    if (!projectFound) {
```

```
                        Toast.makeText(getContext(), "No project found for the
logged-in Team Leader", Toast.LENGTH_SHORT).show();
                    }
                }

                @Override
                public void onCancelled(DatabaseError databaseError) {
                    // Handle any database errors
                    Toast.makeText(getContext(), "Database error: " +
databaseError.getMessage(), Toast.LENGTH_SHORT).show();
                }
            });
        } else {
            Toast.makeText(getContext(), "User not logged in",
Toast.LENGTH_SHORT).show();
        }
    }
}
```