Mobile Computing Lab

Subject Code: MCAL34

A Practical Journal Submitted in Fulfilment of the Degree

Of

MASTER

In

COMPUTER APPLICATION

Year 2024-2025

By

Mr. Agrawal Yash Gopal

(Application Id: - 53715)

Semester- III (CBCS)

Institute of Distance and Open Learning
Vidya Nagari, Kalina, Santacruz East – 400098.
University of Mumbai

PCP Centre
[Vidyavardhini's College of Technology – Vasai Road, Palghar 401202]

Institute of Distance and Open Learning,

Vidya Nagari, Kalina, Santacruz (E) -400098

CERTIFICATE

This to certify that, **Mr. Agrawal Yash Gopal** appearing **Master in Computer Application (Semester III - CBCS) Application ID: 53715** has satisfactorily completed the prescribed practical of **MCAL34 - Mobile Computing Lab** as laid down by the University of Mumbai for the academic year 2024-25.

Teacher in charge	Examiners	Coordinator IDOL, MCA University of Mumbai

Date: -11/01/2025

Place: - Vasai

Index

Sr. No.	Title	Signature
1.	Creating Android Application for Generating User Interface for Student Registration and Feedback Form by using all basic UI controls.	
2.	Creating Android Program to Perform CRUD Operation on SQLite Database.	
3.	Calculator	
4.	To-Do List	
5.	Develop a simple Note Application where Users can Write, Save and Delete Notes.	
6.	Currency Converter and Temperature Converter.	
7.	Create a Quiz Application with Multiple-Choice Questions. You can Add Features like Score Tracking and Different Categories.	
8.	Number Guessing Game	
9.	Stopwatch	
10.	Theme Application for Wallpaper and Background Color Change.	

Practical 1

AIM: Creating Android Application for Generating User Interface for Student Registration and Feedback Form by using all basic UI controls.

MainActivity.java

```
package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.text.InputType;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    private DatabaseHelper databaseHelper;
    private EditText usernameInput, emailInput, passwordInput;
    private ImageView passwordToggle;
    private Button registerButton;
    private boolean isPasswordVisible = false;
    private TextView loginTextView;

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

        // Initialize views
        usernameInput = findViewById(R.id.usernameInput);
        emailInput = findViewById(R.id.emailInput);
        passwordInput = findViewById(R.id.passwordInput);
        passwordToggle = findViewById(R.id.passwordToggle);
        registerButton = findViewById(R.id.registerButton);
        loginTextView = findViewById(R.id.loginTextView);

        databaseHelper = new DatabaseHelper(this);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        loginTextView.setOnClickListener(v -> {
            Intent intent = new Intent(MainActivity.this, LoginActivity.class);
            startActivity(intent);
        });

        // Set up the password visibility toggle
        passwordToggle.setOnClickListener(v -> {
            if (isPasswordVisible) {
                passwordInput.setInputType(InputType.TYPE_CLASS_TEXT                    |
InputType.TYPE_TEXT_VARIATION_PASSWORD);
                passwordToggle.setImageResource(R.drawable.baseline_remove_red_eye_24); // Set the eye icon
to closed
            } else {
                passwordInput.setInputType(InputType.TYPE_CLASS_TEXT                    |
InputType.TYPE_TEXT_VARIATION_VISIBLE_PASSWORD);
                passwordToggle.setImageResource(R.drawable.baseline_remove_red_eye_24); // Set the eye icon
to open
            }
            isPasswordVisible = !isPasswordVisible;
            passwordInput.setSelection(passwordInput.getText().length()); // Move cursor to the end
        });

        registerButton.setOnClickListener(v -> registerUser());
    }

    private void registerUser() {
        String username = usernameInput.getText().toString().trim();
        String email = emailInput.getText().toString().trim();
        String password = passwordInput.getText().toString().trim();

        // Input validation
        if (TextUtils.isEmpty(username)) {
            usernameInput.setError("Username is required");
            return;
        }
        if (TextUtils.isEmpty(email)) {
            emailInput.setError("Email is required");
            return;
        }
        if (TextUtils.isEmpty(password)) {
            passwordInput.setError("Password is required");
            return;
        }
        // Check if email already exists
        if (databaseHelper.checkEmail(email)) {
            Toast.makeText(this, "Email already registered", Toast.LENGTH_SHORT).show();
            return;
        }

        // Check if username already exists
        if (databaseHelper.checkUsername(username)) {
            Toast.makeText(this, "Username already registered", Toast.LENGTH_SHORT).show();
            return;
        }

        // Insert user data into the database
        boolean isInserted = databaseHelper.insertData(username, email, password);
        if (isInserted) {
            Toast.makeText(this, "Registration Successful", Toast.LENGTH_SHORT).show();
            // Redirect to LoginActivity
            Intent intent = new Intent(MainActivity.this, LoginActivity.class);
            startActivity(intent);
            finish(); // Finish MainActivity
        } else {
            Toast.makeText(this, "Registration Failed", Toast.LENGTH_SHORT).show();
        }
    }
}
```

LoginActivity.java

```
package com.example.myapplication;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.text.InputType;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class LoginActivity extends AppCompatActivity {

    private DatabaseHelper databaseHelper;
    private EditText usernameInput, passwordInput;
    private ImageView passwordToggle;
    private Button loginButton;
    private boolean isPasswordVisible = false;
    private TextView click_here_text;

    private static final String PREFS_NAME = "UserPrefs";
    private static final String IS_LOGGED_IN = "isLoggedIn";

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

        // Check if user is already logged in
        SharedPreferences preferences = getSharedPreferences(PREFS_NAME, MODE_PRIVATE);
        boolean isLoggedIn = preferences.getBoolean(IS_LOGGED_IN, false);
        if (isLoggedIn) {
            // Redirect to HomeActivity if logged in
            Intent intent = new Intent(LoginActivity.this, HomeActivity.class);
            startActivity(intent);
            finish(); // Finish LoginActivity
            return; // Exit onCreate
        }

        // Initialize views
        usernameInput = findViewById(R.id.username_edit_text);
        passwordInput = findViewById(R.id.password_edit_text);
        passwordToggle = findViewById(R.id.passwordToggle);
        loginButton = findViewById(R.id.login_button);
        click_here_text = findViewById(R.id.click_here_text);

        databaseHelper = new DatabaseHelper(this);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        click_here_text.setOnClickListener(v -> {
            Intent intent = new Intent(LoginActivity.this, MainActivity.class);
            startActivity(intent);
        });

        // Set up the password visibility toggle
        passwordToggle.setOnClickListener(v -> {
            if (isPasswordVisible) {
                passwordInput.setInputType(InputType.TYPE_CLASS_TEXT                                   |
InputType.TYPE_TEXT_VARIATION_PASSWORD);
                passwordToggle.setImageResource(R.drawable.baseline_remove_red_eye_24); // Set the eye icon
to closed
            } else {
                passwordInput.setInputType(InputType.TYPE_CLASS_TEXT                                    |
InputType.TYPE_TEXT_VARIATION_VISIBLE_PASSWORD);
                passwordToggle.setImageResource(R.drawable.baseline_remove_red_eye_24); // Set the eye icon
to open
            }
            isPasswordVisible = !isPasswordVisible;
            passwordInput.setSelection(passwordInput.getText().length()); // Move cursor to the end
        });

        loginButton.setOnClickListener(v -> loginUser());
    }

    private void loginUser() {
        String usernameOrEmail = usernameInput.getText().toString().trim();
        String password = passwordInput.getText().toString().trim();

        // Input validation
        if (TextUtils.isEmpty(usernameOrEmail)) {
            usernameInput.setError("Username or Email is required");
            return;
        }
        if (TextUtils.isEmpty(password)) {
            passwordInput.setError("Password is required");
            return;
        }

        // Check if username or email exists and matches the password
        if (databaseHelper.checkEmailAndPassword(usernameOrEmail, password) ||
                databaseHelper.checkUsernameAndPassword(usernameOrEmail, password)) {
            Toast.makeText(this, "Login Successful", Toast.LENGTH_SHORT).show();

            // Save login status
            SharedPreferences preferences = getSharedPreferences(PREFS_NAME, MODE_PRIVATE);
            SharedPreferences.Editor editor = preferences.edit();
            editor.putBoolean(IS_LOGGED_IN, true);
            editor.apply();

            // Redirect to HomeActivity
            Intent intent = new Intent(LoginActivity.this, HomeActivity.class);
            startActivity(intent);
            finish(); // Finish the LoginActivity
        } else {
            Toast.makeText(this, "Invalid Username/Email or Password", Toast.LENGTH_SHORT).show();
        }
    }
}
```

HomeActivity.java

```
package com.example.myapplication;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.Button;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class HomeActivity extends AppCompatActivity {

    private static final String PREFS_NAME = "UserPrefs";
    private static final String IS_LOGGED_IN = "isLoggedIn";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_home);

        // Check if user is logged in
        SharedPreferences preferences = getSharedPreferences(PREFS_NAME, MODE_PRIVATE);
        boolean isLoggedIn = preferences.getBoolean(IS_LOGGED_IN, false);
        if (!isLoggedIn) {
            // Redirect to LoginActivity if not logged in
            Intent loginIntent = new Intent(HomeActivity.this, LoginActivity.class);
            startActivity(loginIntent);
            finish(); // Close HomeActivity
            return; // Exit onCreate
        }

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        Button feedbackButton = findViewById(R.id.feedback_button);
        feedbackButton.setOnClickListener(v -> {
            // Handle feedback form action
            Intent feedbackIntent = new Intent(HomeActivity.this, FeedbackActivity.class);
            startActivity(feedbackIntent);
        });

        // Set up the logout button
        Button logoutButton = findViewById(R.id.logout_button);
        logoutButton.setOnClickListener(v -> {
            // Handle logout action (e.g., clear session data and go back to LoginActivity)
            logoutUser();
        });
    }

    private void logoutUser() {
        // Clear user session data
        SharedPreferences sharedPreferences = getSharedPreferences(PREFS_NAME, MODE_PRIVATE);
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.clear(); // Clear all saved data
        editor.apply(); // Apply changes

        // Redirect to LoginActivity
        Intent logoutIntent = new Intent(HomeActivity.this, LoginActivity.class);
        startActivity(logoutIntent);
        finish(); // Close the HomeActivity
    }
}
```

FeedbackActivity.java

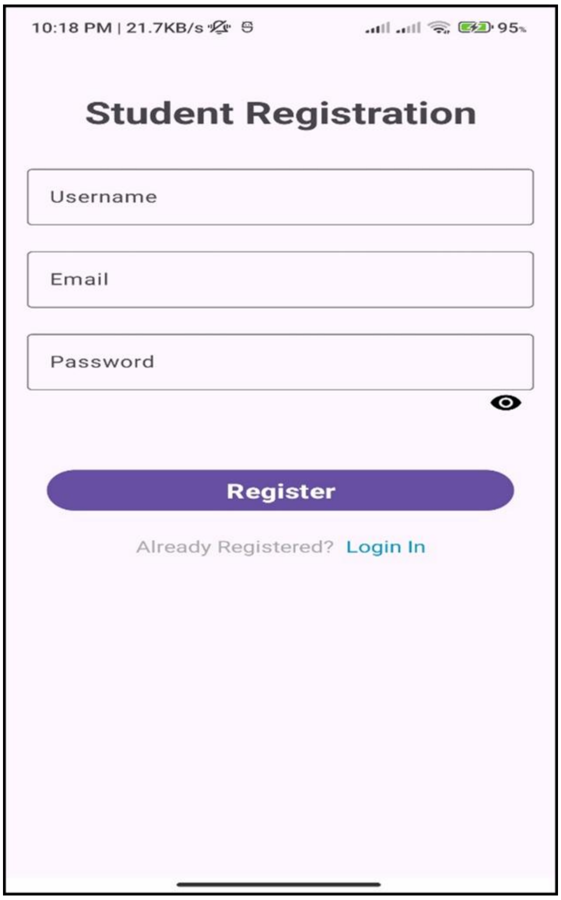
```
package com.example.myapplication;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import com.google.android.material.textfield.TextInputEditText;

public class FeedbackActivity extends AppCompatActivity {

    private TextInputEditText feedbackEditText;
    private DatabaseHelper databaseHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_feedback);

        // Initialize DatabaseHelper
        databaseHelper = new DatabaseHelper(this);
        // Get reference to the feedback input field and button
        feedbackEditText = findViewById(R.id.feedbackEditText);
        Button submitButton = findViewById(R.id.submitButton);

        // Set up window insets for edge-to-edge
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        // Set onClick listener for the submit button
        submitButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                submitFeedback();
            }
        });
    }

    private void submitFeedback() {
        String feedback = feedbackEditText.getText().toString().trim();

        if (feedback.isEmpty()) {
            Toast.makeText(this, "Please fill in the feedback", Toast.LENGTH_SHORT).show();
            return;
        }

        // Insert feedback into the database directly
        // Assuming a default user ID of 1 for demonstration. Adjust this as needed.
        int userId = 1; // You may need to update how you handle user IDs
        if (databaseHelper.insertFeedback(userId, feedback)) {
            Toast.makeText(this, "Feedback submitted successfully", Toast.LENGTH_SHORT).show();
            feedbackEditText.setText(""); // Clear the feedback field
        } else {
            Toast.makeText(this, "Failed to submit feedback", Toast.LENGTH_SHORT).show();
        }
    }
}
```

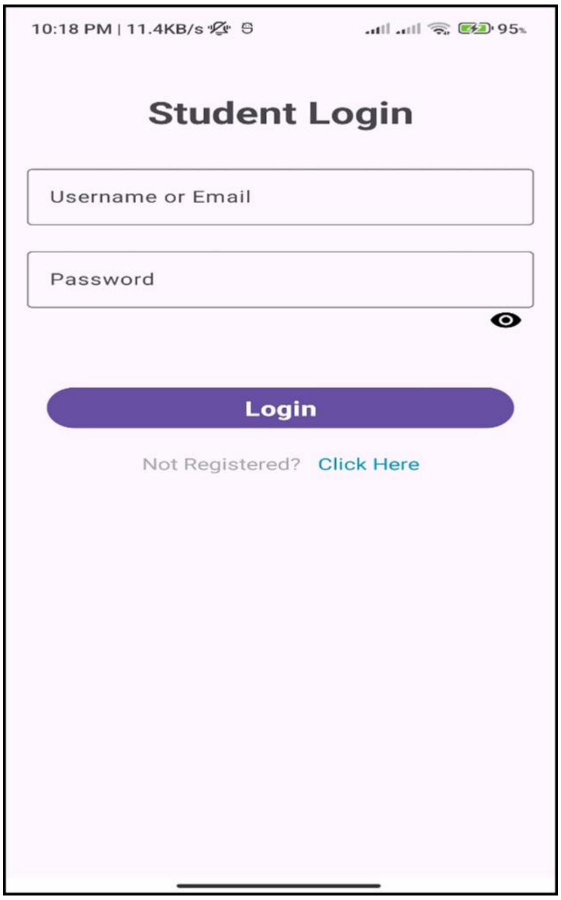
DatabaseHelper.java

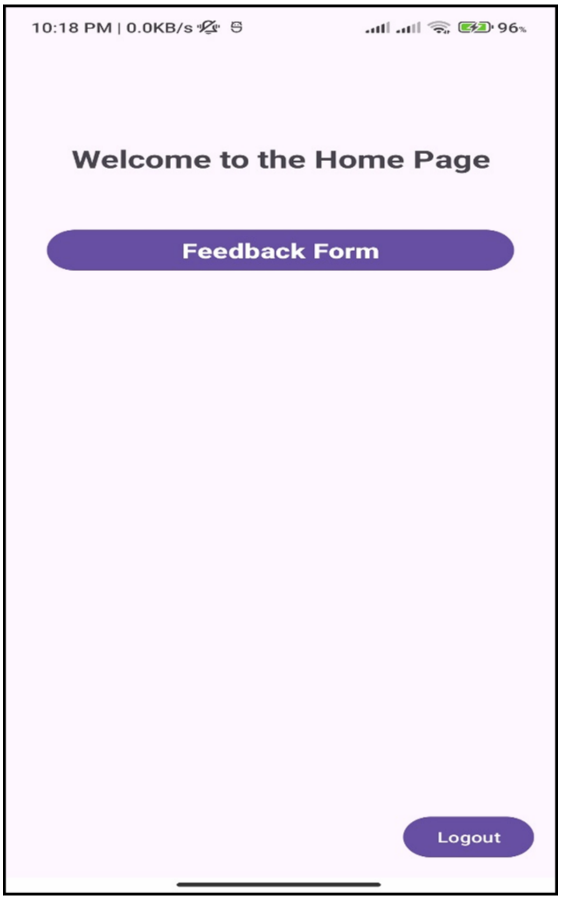
```
package com.example.myapplication;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
```

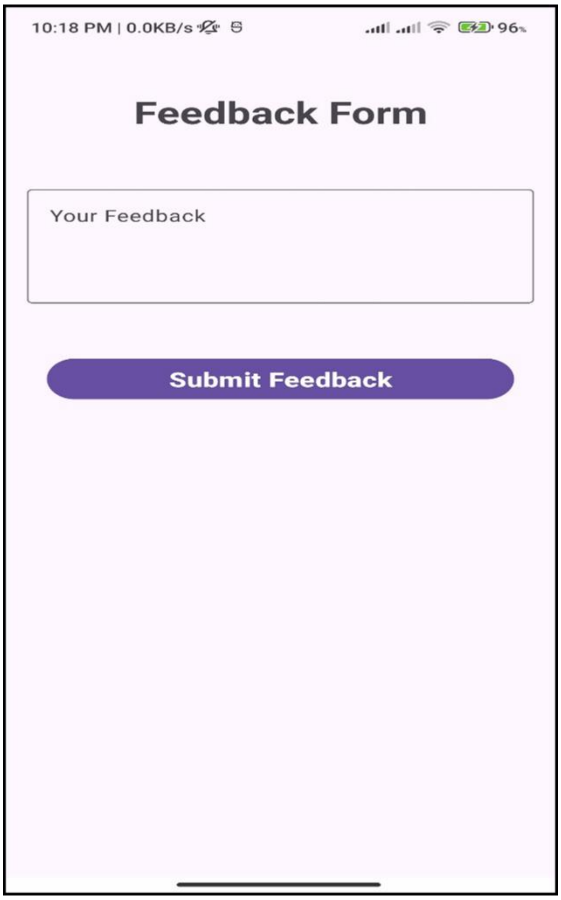
```
public class DatabaseHelper extends SQLiteOpenHelper {

    private static final String DATABASE_NAME = "user_database";
    private static final String TABLE_NAME = "user_table";
    private static final String FEEDBACK_TABLE_NAME = "feedback_table"; // New feedback table
    private static final String COL_1 = "ID";
    private static final String COL_2 = "USERNAME";
    private static final String COL_3 = "EMAIL";
    private static final String COL_4 = "PASSWORD";
    private static final String COL_5 = "FEEDBACK_ID"; // Feedback ID for future reference
    private static final String COL_6 = "USER_ID"; // Foreign key for user
    private static final String COL_7 = "FEEDBACK"; // Feedback content

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, 2); // Increment version number if schema changes
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // Create user table
        db.execSQL("CREATE TABLE " + TABLE_NAME + " (ID INTEGER PRIMARY KEY AUTOINCREMENT,
USERNAME TEXT, EMAIL TEXT, PASSWORD TEXT)");

        // Create feedback table
        db.execSQL("CREATE TABLE " + FEEDBACK_TABLE_NAME + " (FEEDBACK_ID INTEGER PRIMARY KEY
AUTOINCREMENT, USER_ID INTEGER, FEEDBACK TEXT, FOREIGN KEY (USER_ID) REFERENCES " +
TABLE_NAME + " (ID))");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + FEEDBACK_TABLE_NAME);
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }

    public boolean insertData(String username, String email, String password) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_2, username);
        contentValues.put(COL_3, email);
        contentValues.put(COL_4, password);
        long result = db.insert(TABLE_NAME, null, contentValues);
        return result != -1; // Returns true if the data was inserted successfully
    }

    public boolean insertFeedback(int userId, String feedback) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_6, userId); // Store user ID with feedback
        contentValues.put(COL_7, feedback);
        long result = db.insert(FEEDBACK_TABLE_NAME, null, contentValues);
        if (result == -1) {
            Log.e("DatabaseError", "Failed to insert feedback: " + feedback);
        }
        return result != -1; // Returns true if the feedback was inserted successfully
    }

    public boolean checkEmailAndPassword(String email, String password) {
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME + " WHERE EMAIL=? AND
PASSWORD=?", new String[]{email, password});
        return cursor.getCount() > 0; // Returns true if the email and password match
    }

    public boolean checkUsernameAndPassword(String username, String password) {
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME + " WHERE USERNAME=? AND
PASSWORD=?", new String[]{username, password});
        return cursor.getCount() > 0; // Returns true if the username and password match
    }

    public boolean checkEmail(String email) {
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME + " WHERE EMAIL=?", new
String[]{email});
        return cursor.getCount() > 0; // Returns true if the email exists
    }

    public boolean checkUsername(String username) {
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME + " WHERE USERNAME=?", new
String[]{username});
        return cursor.getCount() > 0; // Returns true if the username exists
    }
}
```

OUTPUT:


10:18 PM | 21.7KB/s
95%
Student Registration
Username
Email
Password
Register
Already Registered? Login In


10:18 PM | 11.4KB/s
95%
Student Login
Username or Email
Password
Login
Not Registered? Click Here


10:18 PM | 0.0KB/s
96%
Welcome to the Home Page
Feedback Form
Logout


10:18 PM | 0.0KB/s
96%
Feedback Form
Your Feedback
Submit Feedback

Practical 2

AIM: Creating Android Program to Perform CRUD Operation on SQLite Database.

AddTaskActivity.java

```
package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast; // Import Toast for displaying messages
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import com.google.android.material.textfield.TextInputEditText;

public class AddTaskActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_add_task);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        Button doneButton = findViewById(R.id.doneButton);
        doneButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Retrieve task name and description
                String taskName = ((TextInputEditText) findViewById(R.id.taskNameEditText)).getText().toString();
                String taskDescription = ((TextInputEditText)
findViewById(R.id.taskDescriptionEditText)).getText().toString();

                // Check if fields are empty
                if (taskName.isEmpty() || taskDescription.isEmpty()) {
                    // Show a toast message to inform the user
                    Toast.makeText(AddTaskActivity.this, "Please fill in all fields", Toast.LENGTH_SHORT).show();
                } else {
                    // Insert into database
                    DatabaseHelper dbHelper = new DatabaseHelper(AddTaskActivity.this);
                    dbHelper.insertTask(taskName, taskDescription);
                    // Return to MainActivity
                    Intent intent = new Intent(AddTaskActivity.this, MainActivity.class);
                    startActivity(intent);
                    finish();
                }
            }
        });
    }
}
```

DatabaseHelper.java

```
package com.example.myapplication;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import java.util.ArrayList;

public class DatabaseHelper extends SQLiteOpenHelper {

    private static final String DATABASE_NAME = "tasks.db";
    private static final String TABLE_NAME = "tasks";
    private static final String COL_ID = "id";
    private static final String COL_NAME = "name";
    private static final String COL_DESCRIPTION = "description";

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTable = "CREATE TABLE " + TABLE_NAME + " (" +
                COL_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
                COL_NAME + " TEXT, " +
                COL_DESCRIPTION + " TEXT)";
        db.execSQL(createTable);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }

    public boolean insertTask(String name, String description) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_NAME, name);
        contentValues.put(COL_DESCRIPTION, description);
        long result = db.insert(TABLE_NAME, null, contentValues);
        return result != -1;
    }

    public ArrayList<Task> getAllTasksWithDetails() {
        ArrayList<Task> taskList = new ArrayList<>();
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
        if (cursor.moveToFirst()) {
            do {
                int id = cursor.getInt(0);
                String name = cursor.getString(1);
                String description = cursor.getString(2);
                taskList.add(new Task(id, name, description));
            } while (cursor.moveToNext());
        }
        cursor.close();
        return taskList;
    }

    public void deleteTask(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_NAME, COL_ID + " = ?", new String[]{String.valueOf(id)});
    }

    public boolean updateTask(int id, String name, String description) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_NAME, name);
        contentValues.put(COL_DESCRIPTION, description);
        int result = db.update(TABLE_NAME, contentValues, COL_ID + " = ?", new String[]{String.valueOf(id)});
        return result > 0;
    }
}
```

EditTaskActivity.java

```
package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class EditTaskActivity extends AppCompatActivity {

    private EditText taskNameEditText;
    private EditText taskDescriptionEditText;
    private DatabaseHelper dbHelper;
    private int taskId;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_task);
        EdgeToEdge.enable(this);

        taskNameEditText = findViewById(R.id.taskNameEditText);
        taskDescriptionEditText = findViewById(R.id.taskDescriptionEditText);
        Button updateButton = findViewById(R.id.updateButton);
        dbHelper = new DatabaseHelper(this);

        Intent intent = getIntent();
        taskId = intent.getIntExtra("TASK_ID", -1);
        String taskName = intent.getStringExtra("TASK_NAME");
        String taskDescription = intent.getStringExtra("TASK_DESCRIPTION");

        taskNameEditText.setText(taskName);
        taskDescriptionEditText.setText(taskDescription);

        updateButton.setOnClickListener(v -> {
            String name = taskNameEditText.getText().toString();
            String description = taskDescriptionEditText.getText().toString();

            // Check if fields are empty
            if (name.isEmpty() || description.isEmpty()) {
                // Show a toast message to inform the user
                Toast.makeText(EditTaskActivity.this, "Please fill in all fields", Toast.LENGTH_SHORT).show();
            } else {
                // Proceed with the update
                if (dbHelper.updateTask(taskId, name, description)) {
                    Toast.makeText(EditTaskActivity.this, "Note updated successfully",
Toast.LENGTH_SHORT).show();
                    finish();
                } else {
                    Toast.makeText(EditTaskActivity.this, "Failed to update Note.", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

MainActivity.java

```
package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    private ListView taskListView;
    private DatabaseHelper dbHelper;
    private TaskAdapter adapter;
    private ArrayList<Task> tasks;

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

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        // Initialize database helper
        dbHelper = new DatabaseHelper(this);
        tasks = new ArrayList<>();

        // Set up ListView
        taskListView = findViewById(R.id.taskListView);
        displayTasks();

        // Set up the "Add Task" button
        Button addButton = findViewById(R.id.addButton);
        addButton.setOnClickListener(v -> {
            Intent intent = new Intent(MainActivity.this, AddTaskActivity.class);
            startActivity(intent);
        });
    }

    private void displayTasks() {
        tasks = dbHelper.getAllTasksWithDetails();
        adapter = new TaskAdapter(this, tasks, dbHelper);
        taskListView.setAdapter(adapter);
    }

    @Override
    protected void onResume() {
        super.onResume();
        displayTasks(); // Refresh task list when returning to this activity
    }
}
```

Task.java

```
package com.example.myapplication;

public class Task {
    private int id;
    private String name;
    private String description;

    public Task(int id, String name, String description) {
        this.id = id;
        this.name = name;
        this.description = description;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public String getDescription() {
        return description;
    }
}
```

TaskAdapter.java

```
package com.example.myapplication;
```

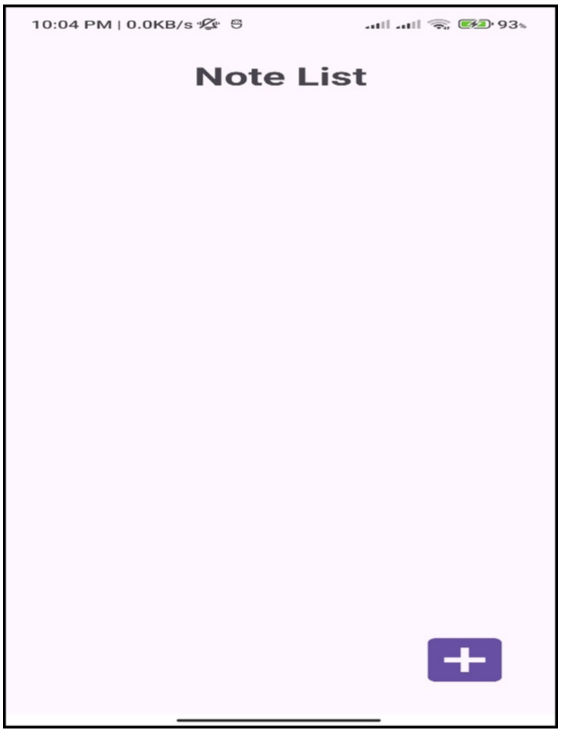
```
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.TextView;

import java.util.ArrayList;

public class TaskAdapter extends ArrayAdapter<Task> {
   private final Context context;
   private final ArrayList<Task> tasks;
   private final DatabaseHelper dbHelper;

   public TaskAdapter(Context context, ArrayList<Task> tasks, DatabaseHelper dbHelper) {
      super(context, R.layout.list_item_task, tasks);
      this.context = context;
      this.tasks = tasks;
      this.dbHelper = dbHelper;
   }

   @Override
   public View getView(int position, View convertView, ViewGroup parent) {
      LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
      View rowView = inflater.inflate(R.layout.list_item_task, parent, false);

      TextView taskNameTextView = rowView.findViewById(R.id.taskNameTextView);
      Button editButton = rowView.findViewById(R.id.editButton);
      Button viewButton = rowView.findViewById(R.id.viewButton);
      Button deleteButton = rowView.findViewById(R.id.deleteButton);

      Task task = tasks.get(position);
      taskNameTextView.setText(task.getName());

      viewButton.setOnClickListener(v -> {
         Intent intent = new Intent(context, ViewTaskActivity.class);
         intent.putExtra("TASK_ID", task.getId());
         intent.putExtra("TASK_NAME", task.getName());
         intent.putExtra("TASK_DESCRIPTION", task.getDescription());
         context.startActivity(intent);
      });

      deleteButton.setOnClickListener(v -> {
         dbHelper.deleteTask(task.getId());
         tasks.remove(position);
         notifyDataSetChanged();
      });
```

```
        editButton.setOnClickListener(v -> {
            Intent intent = new Intent(context, EditTaskActivity.class);
            intent.putExtra("TASK_ID", task.getId());
            intent.putExtra("TASK_NAME", task.getName());
            intent.putExtra("TASK_DESCRIPTION", task.getDescription());
            context.startActivity(intent);
        });

        return rowView;
    }
}
```

ViewTaskActivity.java

```
package com.example.myapplication;

import android.os.Bundle;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;

public class ViewTaskActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_task);
        EdgeToEdge.enable(this);

        // Retrieve data from the intent
        int taskId = getIntent().getIntExtra("TASK_ID", -1);
        String taskName = getIntent().getStringExtra("TASK_NAME");
        String taskDescription = getIntent().getStringExtra("TASK_DESCRIPTION");

        // Find views and set data
        TextView taskNameTextView = findViewById(R.id.taskNameTextView);
        TextView taskDescriptionTextView = findViewById(R.id.taskDescriptionTextView);

        taskNameTextView.setText(taskName);
        taskDescriptionTextView.setText(taskDescription);
    }
}
```

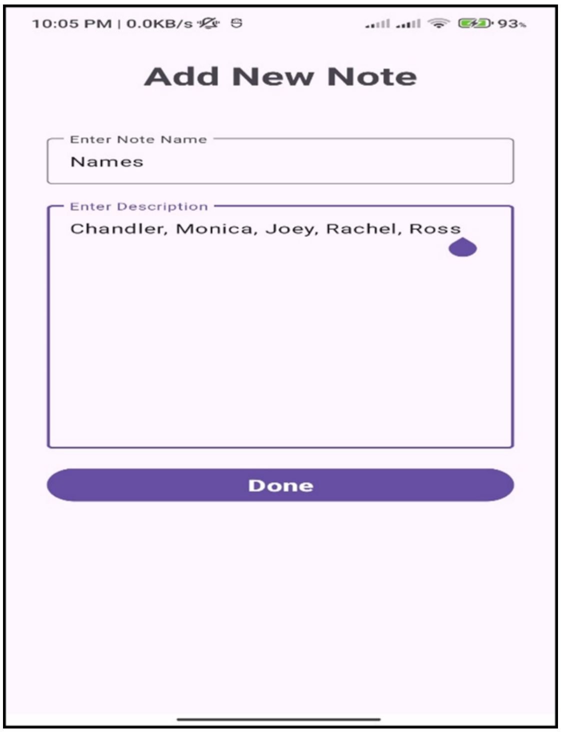
OUTPUT:



10:04 PM | 0.0KB/s 93%

Note List

+



10:05 PM | 0.0KB/s 93%

Add New Note

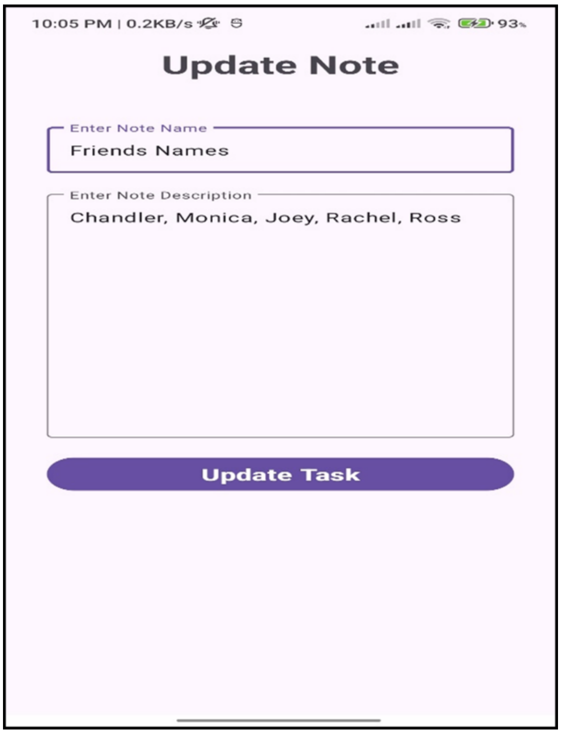
Enter Note Name

Names

Enter Description

Chandler, Monica, Joey, Rachel, Ross

Done



10:05 PM | 0.2KB/s 93%

Update Note

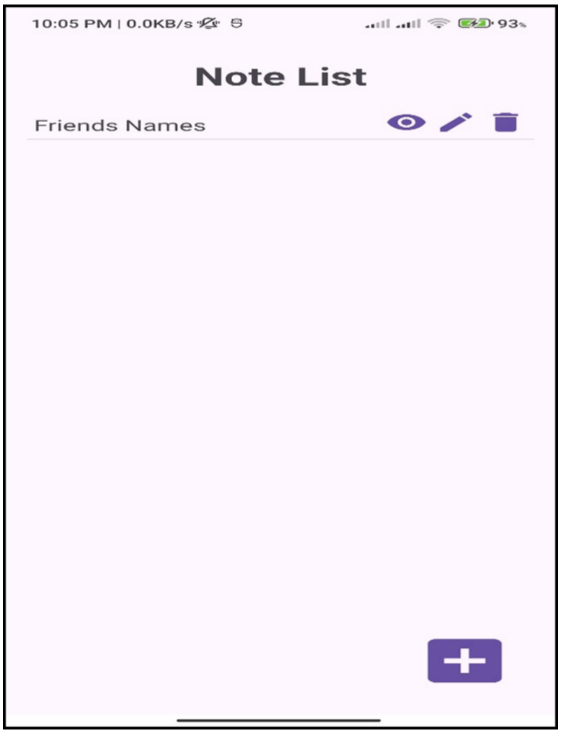
Enter Note Name

Friends Names

Enter Note Description

Chandler, Monica, Joey, Rachel, Ross

Update Task



10:05 PM | 0.0KB/s 93%

Note List

Friends Names

+

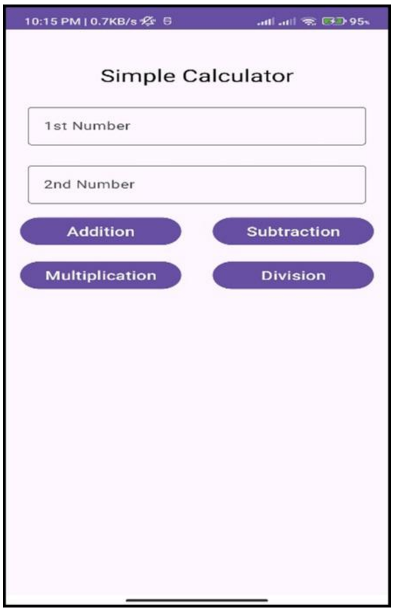
Practical 3

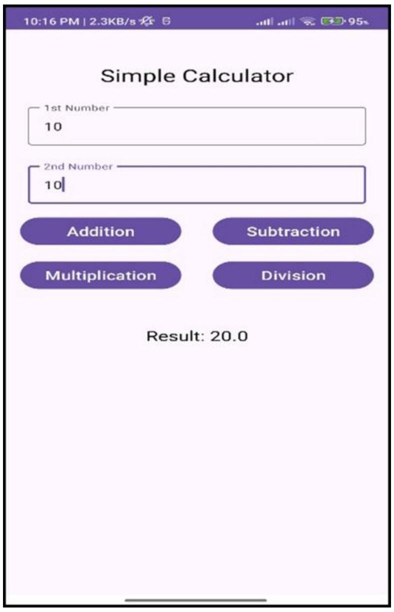
AIM: Calculator

MainActivity.java

```
package com.example.myapplication;

import android.os.Bundle;
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.material.textfield.TextInputEditText;

public class MainActivity extends AppCompatActivity {

    private TextInputEditText input1, input2;
    private TextView resultTextView;

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

        // Initialize views
        input1 = findViewById(R.id.textInputEditText1);
        input2 = findViewById(R.id.textInputEditText2);
        resultTextView = findViewById(R.id.bottomTextView);

        // Buttons
        Button addButton = findViewById(R.id.button1);
        Button subtractButton = findViewById(R.id.button2);
        Button multiplyButton = findViewById(R.id.button3);
        Button divideButton = findViewById(R.id.button4);

        // Set click listeners for each button
        addButton.setOnClickListener(v -> performOperation(Operation.ADD));
        subtractButton.setOnClickListener(v -> performOperation(Operation.SUBTRACT));
        multiplyButton.setOnClickListener(v -> performOperation(Operation.MULTIPLY));
        divideButton.setOnClickListener(v -> performOperation(Operation.DIVIDE));
    }

    // Method to perform the selected operation
    private void performOperation(Operation operation) {
        String inputText1 = input1.getText().toString();
        String inputText2 = input2.getText().toString();

        if (inputText1.isEmpty() || inputText2.isEmpty()) {
            Toast.makeText(this, "Please enter both numbers", Toast.LENGTH_SHORT).show();
            return;
        }

        try {
            double num1 = Double.parseDouble(inputText1);
            double num2 = Double.parseDouble(inputText2);
            double result = 0;

            // Perform the operation
            switch (operation) {
                case ADD:
                    result = num1 + num2;
                    break;
                case SUBTRACT:
                    result = num1 - num2;
                    break;
                case MULTIPLY:
                    result = num1 * num2;
                    break;
                case DIVIDE:
                    if (num2 != 0) {
                        result = num1 / num2;
                    } else {
                        Toast.makeText(this, "Cannot divide by zero", Toast.LENGTH_SHORT).show();
                        return;
                    }
                    break;
            }

            // Display the result
            resultTextView.setText("Result: " + result);

        } catch (NumberFormatException e) {
            Toast.makeText(this, "Invalid number format", Toast.LENGTH_SHORT).show();
        }
    }

    // Enum for the operations
    private enum Operation {
        ADD, SUBTRACT, MULTIPLY, DIVIDE
    }
}
```

OUTPUT:





1st Number: 10

2nd Number: 10

Result: 20.0

Practical 4

AIM: To-Do List

DatabaseHelper.java

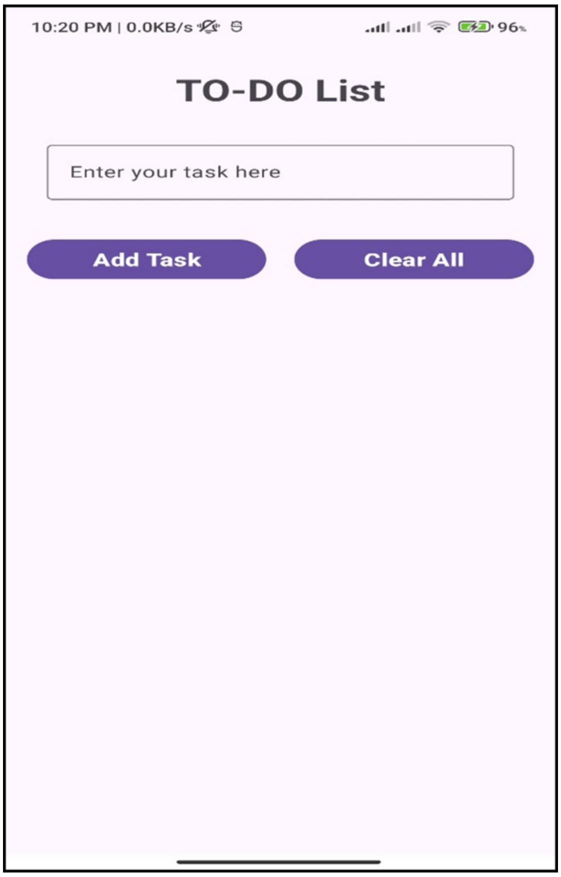
```
package com.example.myapplication;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import java.util.ArrayList;

public class DatabaseHelper extends SQLiteOpenHelper {
   private static final String DATABASE_NAME = "todo.db";
   private static final int DATABASE_VERSION = 1;
   private static final String TABLE_TODOS = "todos";
   private static final String COLUMN_ID = "_id";
   private static final String COLUMN_TASK = "task";

   public DatabaseHelper(Context context) {
      super(context, DATABASE_NAME, null, DATABASE_VERSION);
   }

   @Override
   public void onCreate(SQLiteDatabase db) {
      String createTable = "CREATE TABLE " + TABLE_TODOS + " (" +
            COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
            COLUMN_TASK + " TEXT)";
      db.execSQL(createTable);

      // Create index on task column
      String createIndex = "CREATE INDEX idx_task ON " + TABLE_TODOS + " (" + COLUMN_TASK + ")";
      db.execSQL(createIndex);
   }

   @Override
   public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
      db.execSQL("DROP TABLE IF EXISTS " + TABLE_TODOS);
      db.execSQL("DROP INDEX IF EXISTS idx_task");
      onCreate(db);
   }

   // Method to add a task
   public void addTask(String task) {
      SQLiteDatabase db = this.getWritableDatabase();
      ContentValues values = new ContentValues();
      values.put(COLUMN_TASK, task);
```

```
        db.insert(TABLE_TODOS, null, values);
        db.close();
    }

    // Method to get all tasks
    public ArrayList<String> getAllTasks() {
        ArrayList<String> tasks = new ArrayList<>();
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_TODOS, null);
        if (cursor.moveToFirst()) {
            do {
                tasks.add(cursor.getString(cursor.getColumnIndex(COLUMN_TASK)));
            } while (cursor.moveToNext());
        }
        cursor.close();
        db.close();
        return tasks;
    }

    // Method to clear all tasks
    public void clearTasks() {
        SQLiteDatabase db = this.getWritableDatabase();
        db.execSQL("DELETE FROM " + TABLE_TODOS);
        db.close();
    }
}
```

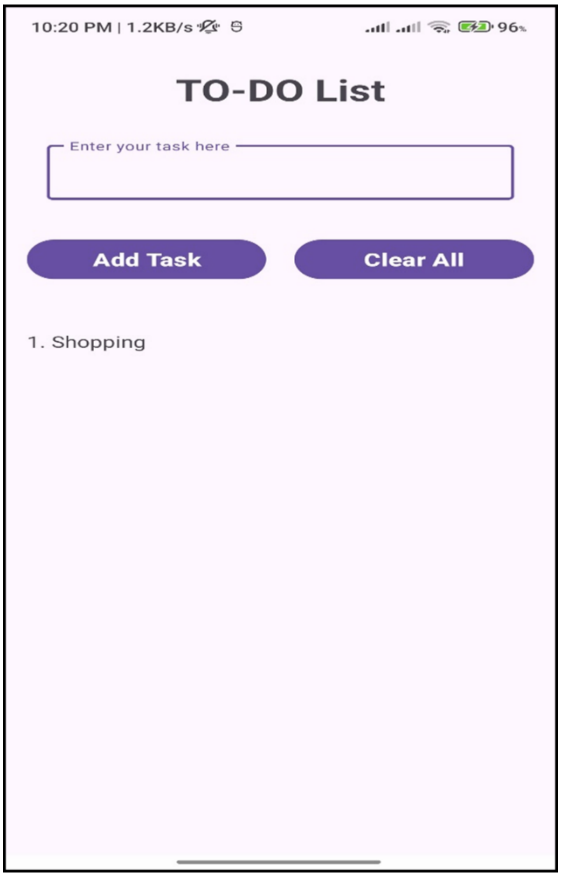
MainActivity.java

```
package com.example.myapplication;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    private DatabaseHelper dbHelper;
    private EditText inputEditText;
    private TextView tasksTextView;

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

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        dbHelper = new DatabaseHelper(this);
        inputEditText = findViewById(R.id.textInputEditText);
        tasksTextView = findViewById(R.id.textViewTasks);

        Button addButton = findViewById(R.id.buttonAddTask);
        Button clearButton = findViewById(R.id.buttonClearAllTasks);

        addButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String task = inputEditText.getText().toString();
                if (!task.isEmpty()) {
                    dbHelper.addTask(task);
                    inputEditText.setText("");
                    displayTasks();
                }
            }
        });

        clearButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                dbHelper.clearTasks();
                displayTasks();
            }
        });

        // Display existing tasks on startup
        displayTasks();
    }

    private void displayTasks() {
        ArrayList<String> tasks = dbHelper.getAllTasks();
        StringBuilder tasksDisplay = new StringBuilder();
        for (int i = 0; i < tasks.size(); i++) {
            tasksDisplay.append(i + 1).append(". ").append(tasks.get(i)).append("\n");
        }
        tasksTextView.setText(tasksDisplay.toString());
    }
}
```

OUTPUT:



10:20 PM | 0.0KB/s



10:20 PM | 1.2KB/s

1. Shopping

Practical 5

AIM: Develop a simple Note Application where Users can Write, Save and Delete Notes.

AddTaskActivity.java

```
package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast; // Import Toast for displaying messages
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import com.google.android.material.textfield.TextInputEditText;

public class AddTaskActivity extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_add_task);

    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
      Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
      v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
      return insets;
    });

    Button doneButton = findViewById(R.id.doneButton);
    doneButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        // Retrieve task name and description
        String taskName = ((TextInputEditText) findViewById(R.id.taskNameEditText)).getText().toString();
        String taskDescription = ((TextInputEditText)
findViewById(R.id.taskDescriptionEditText)).getText().toString();

        // Check if fields are empty
        if (taskName.isEmpty() || taskDescription.isEmpty()) {
          // Show a toast message to inform the user
          Toast.makeText(AddTaskActivity.this, "Please fill in all fields", Toast.LENGTH_SHORT).show();
        } else {
          // Insert into database
          DatabaseHelper dbHelper = new DatabaseHelper(AddTaskActivity.this);
```

```
                    dbHelper.insertTask(taskName, taskDescription);

                    // Return to MainActivity
                    Intent intent = new Intent(AddTaskActivity.this, MainActivity.class);
                    startActivity(intent);
                    finish();
                }
            }
        });
    }
}
```

DatabaseHelper.java

```
package com.example.myapplication;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import java.util.ArrayList;

public class DatabaseHelper extends SQLiteOpenHelper {

    private static final String DATABASE_NAME = "tasks.db";
    private static final String TABLE_NAME = "tasks";
    private static final String COL_ID = "id";
    private static final String COL_NAME = "name";
    private static final String COL_DESCRIPTION = "description";

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTable = "CREATE TABLE " + TABLE_NAME + " (" +
                COL_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
                COL_NAME + " TEXT, " +
                COL_DESCRIPTION + " TEXT)";
        db.execSQL(createTable);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
    public boolean insertTask(String name, String description) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_NAME, name);
        contentValues.put(COL_DESCRIPTION, description);
        long result = db.insert(TABLE_NAME, null, contentValues);
        return result != -1;
    }

    public ArrayList<Task> getAllTasksWithDetails() {
        ArrayList<Task> taskList = new ArrayList<>();
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
        if (cursor.moveToFirst()) {
            do {
                int id = cursor.getInt(0);
                String name = cursor.getString(1);
                String description = cursor.getString(2);
                taskList.add(new Task(id, name, description));
            } while (cursor.moveToNext());
        }
        cursor.close();
        return taskList;
    }

    public void deleteTask(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_NAME, COL_ID + " = ?", new String[]{String.valueOf(id)});
    }

    public boolean updateTask(int id, String name, String description) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_NAME, name);
        contentValues.put(COL_DESCRIPTION, description);
        int result = db.update(TABLE_NAME, contentValues, COL_ID + " = ?", new String[]{String.valueOf(id)});
        return result > 0;
    }
}
```

EditTaskActivity.java

```
package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class EditTaskActivity extends AppCompatActivity {

    private EditText taskNameEditText;
    private EditText taskDescriptionEditText;
    private DatabaseHelper dbHelper;
    private int taskId;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_task);
        EdgeToEdge.enable(this);

        taskNameEditText = findViewById(R.id.taskNameEditText);
        taskDescriptionEditText = findViewById(R.id.taskDescriptionEditText);
        Button updateButton = findViewById(R.id.updateButton);
        dbHelper = new DatabaseHelper(this);

        Intent intent = getIntent();
        taskId = intent.getIntExtra("TASK_ID", -1);
        String taskName = intent.getStringExtra("TASK_NAME");
        String taskDescription = intent.getStringExtra("TASK_DESCRIPTION");

        taskNameEditText.setText(taskName);
        taskDescriptionEditText.setText(taskDescription);

        updateButton.setOnClickListener(v -> {
            String name = taskNameEditText.getText().toString();
            String description = taskDescriptionEditText.getText().toString();

            // Check if fields are empty
            if (name.isEmpty() || description.isEmpty()) {
                // Show a toast message to inform the user
                Toast.makeText(EditTaskActivity.this, "Please fill in all fields", Toast.LENGTH_SHORT).show();
            } else {
                // Proceed with the update
                if (dbHelper.updateTask(taskId, name, description)) {
                    Toast.makeText(EditTaskActivity.this, "Note updated successfully",
Toast.LENGTH_SHORT).show();
                    finish();
                } else {
                    Toast.makeText(EditTaskActivity.this, "Failed to update Note.", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

MainActivity.java

```
package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    private ListView taskListView;
    private DatabaseHelper dbHelper;
    private TaskAdapter adapter;
    private ArrayList<Task> tasks;

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

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        // Initialize database helper
        dbHelper = new DatabaseHelper(this);
        tasks = new ArrayList<>();

        // Set up ListView
        taskListView = findViewById(R.id.taskListView);
        displayTasks();
        // Set up the "Add Task" button
        Button addButton = findViewById(R.id.addButton);
        addButton.setOnClickListener(v -> {
            Intent intent = new Intent(MainActivity.this, AddTaskActivity.class);
            startActivity(intent);
        });
    }

    private void displayTasks() {
        tasks = dbHelper.getAllTasksWithDetails();
        adapter = new TaskAdapter(this, tasks, dbHelper);
        taskListView.setAdapter(adapter);
    }

    @Override
    protected void onResume() {
        super.onResume();
        displayTasks(); // Refresh task list when returning to this activity
    }
}
```

Task.java

```
package com.example.myapplication;

public class Task {
    private int id;
    private String name;
    private String description;

    public Task(int id, String name, String description) {
        this.id = id;
        this.name = name;
        this.description = description;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public String getDescription() {
        return description;
    }
}
```

TaskAdapter.java

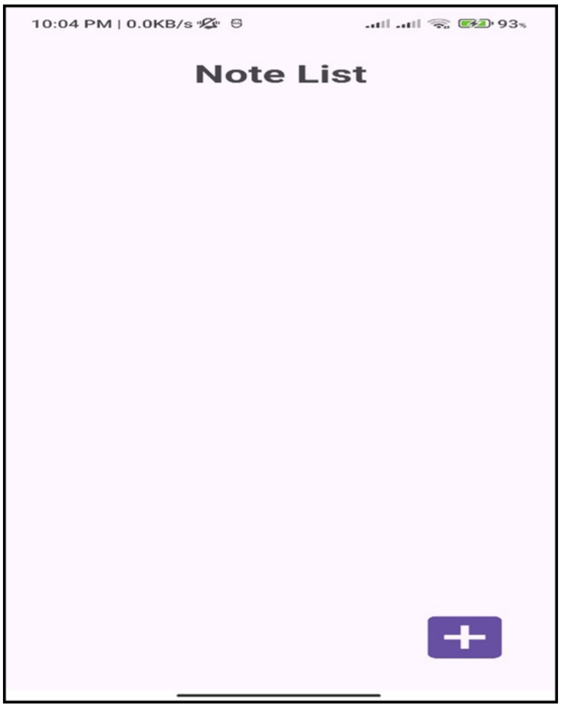
```
package com.example.myapplication;
```

```
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.TextView;

import java.util.ArrayList;

public class TaskAdapter extends ArrayAdapter<Task> {
    private final Context context;
    private final ArrayList<Task> tasks;
    private final DatabaseHelper dbHelper;

    public TaskAdapter(Context context, ArrayList<Task> tasks, DatabaseHelper dbHelper) {
        super(context, R.layout.list_item_task, tasks);
        this.context = context;
        this.tasks = tasks;
        this.dbHelper = dbHelper;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View rowView = inflater.inflate(R.layout.list_item_task, parent, false);

        TextView taskNameTextView = rowView.findViewById(R.id.taskNameTextView);
        Button editButton = rowView.findViewById(R.id.editButton);
        Button viewButton = rowView.findViewById(R.id.viewButton);
        Button deleteButton = rowView.findViewById(R.id.deleteButton);

        Task task = tasks.get(position);
        taskNameTextView.setText(task.getName());

        viewButton.setOnClickListener(v -> {
            Intent intent = new Intent(context, ViewTaskActivity.class);
            intent.putExtra("TASK_ID", task.getId());
            intent.putExtra("TASK_NAME", task.getName());
            intent.putExtra("TASK_DESCRIPTION", task.getDescription());
            context.startActivity(intent);
        });

        deleteButton.setOnClickListener(v -> {
            dbHelper.deleteTask(task.getId());
            tasks.remove(position);
            notifyDataSetChanged();
        });

        editButton.setOnClickListener(v -> {
            Intent intent = new Intent(context, EditTaskActivity.class);
            intent.putExtra("TASK_ID", task.getId());
            intent.putExtra("TASK_NAME", task.getName());
            intent.putExtra("TASK_DESCRIPTION", task.getDescription());
            context.startActivity(intent);
        });

        return rowView;
    }
}
```

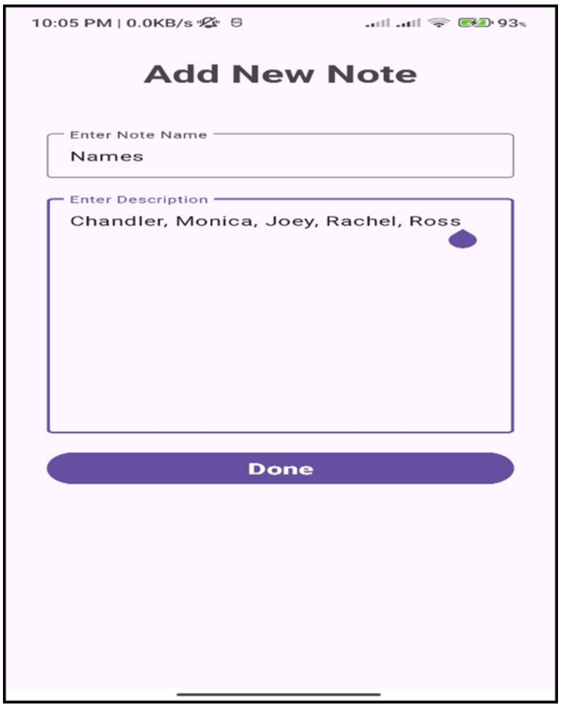
ViewTaskActivity.java

```
package com.example.myapplication;

import android.os.Bundle;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;

public class ViewTaskActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_task);
        EdgeToEdge.enable(this);

        // Retrieve data from the intent
        int taskId = getIntent().getIntExtra("TASK_ID", -1);
        String taskName = getIntent().getStringExtra("TASK_NAME");
        String taskDescription = getIntent().getStringExtra("TASK_DESCRIPTION");

        // Find views and set data
        TextView taskNameTextView = findViewById(R.id.taskNameTextView);
        TextView taskDescriptionTextView = findViewById(R.id.taskDescriptionTextView);

        taskNameTextView.setText(taskName);
        taskDescriptionTextView.setText(taskDescription);
    }
}
```

OUTPUT:



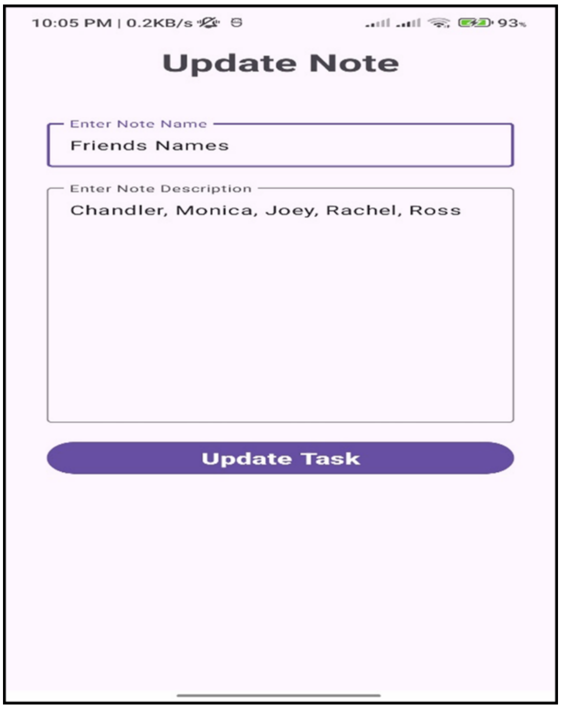
Note List



Add New Note

Enter Note Name: Names

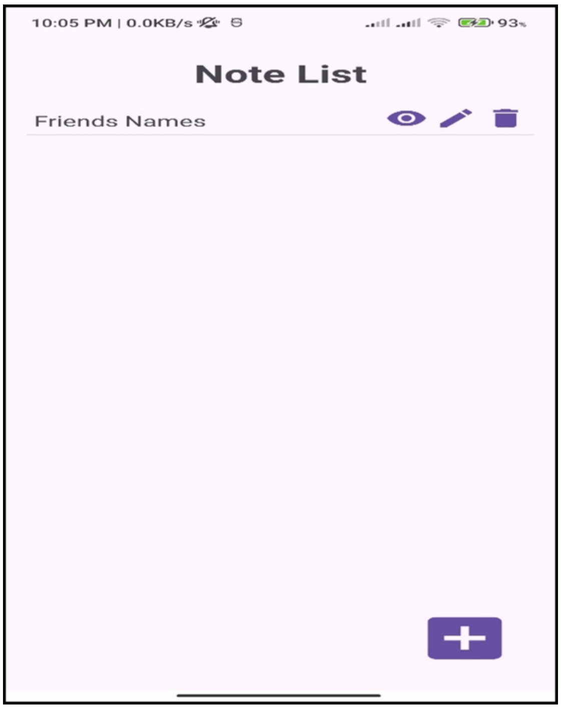
Enter Description: Chandler, Monica, Joey, Rachel, Ross



Update Note

Enter Note Name: Friends Names

Enter Note Description: Chandler, Monica, Joey, Rachel, Ross



Note List

Friends Names

Practical 6

AIM: Currency Converter and Temperature Converter.

CurrencyConverterActivity.java

```
package com.example.myapplication;

import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.TextView;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import com.google.android.material.textfield.TextInputEditText;
import android.widget.Spinner;

public class CurrencyConverterActivity extends AppCompatActivity {

    private TextInputEditText inputAmount;
    private Spinner fromCurrency, toCurrency;
    private Button convertButton;
    private TextView resultText;

    // List of currencies and their country names
    private String[] currencies = {
            "AED - United Arab Emirates Dirham",
            "AFN - Afghan Afghani",
            "ALL - Albanian Lek",
            "AMD - Armenian Dram",
            "ANG - Netherlands Antillean Guilder",
            "AOA - Angolan Kwanza",
            "ARS - Argentine Peso",
            "AUD - Australian Dollar",
            "AWG - Aruban Florin",
            "AZN - Azerbaijani Manat",
            "BAM - Bosnia and Herzegovina Convertible Mark",
            "BBD - Barbadian Dollar",
            "BDT - Bangladeshi Taka",
            "BGN - Bulgarian Lev",
            "BHD - Bahraini Dinar",
            "BIF - Burundian Franc",
            "BMD - Bermudian Dollar",
            "BND - Brunei Dollar",
            "BOB - Bolivian Boliviano",
            "BRL - Brazilian Real",
"BSD - Bahamian Dollar",
"BTN - Bhutanese Ngultrum",
"BWP - Botswana Pula",
"BYN - Belarusian Ruble",
"BZD - Belize Dollar",
"CAD - Canadian Dollar",
"CDF - Congolese Franc",
"CHF - Swiss Franc",
"CLP - Chilean Peso",
"CNY - Chinese Yuan",
"COP - Colombian Peso",
"CRC - Costa Rican Colón",
"CUP - Cuban Peso",
"CVE - Cape Verdean Escudo",
"CZK - Czech Koruna",
"DJF - Djiboutian Franc",
"DKK - Danish Krone",
"DOP - Dominican Peso",
"DZD - Algerian Dinar",
"EGP - Egyptian Pound",
"ERN - Eritrean Nakfa",
"ETB - Ethiopian Birr",
"EUR - Euro",
"FJD - Fijian Dollar",
"FKP - Falkland Islands Pound",
"FOK - Faroese Króna",
"GBP - British Pound Sterling",
"GEL - Georgian Lari",
"GGP - Guernsey Pound",
"GHS - Ghanaian Cedi",
"GIP - Gibraltar Pound",
"GMD - Gambian Dalasi",
"GNF - Guinean Franc",
"GTQ - Guatemalan Quetzal",
"GYD - Guyanese Dollar",
"HKD - Hong Kong Dollar",
"HNL - Honduran Lempira",
"HRK - Croatian Kuna",
"HTG - Haitian Gourde",
"HUF - Hungarian Forint",
"IDR - Indonesian Rupiah",
"ILS - Israeli New Shekel",
"IMP - Isle of Man Pound",
"INR - Indian Rupee",
"IQD - Iraqi Dinar",
"IRR - Iranian Rial",
"ISK - Icelandic Króna",
"JEP - Jersey Pound",
"JMD - Jamaican Dollar",
"JPY - Japanese Yen",
    "KES - Kenyan Shilling",
    "KGS - Kyrgyzstani Som",
    "KHR - Cambodian Riel",
    "KPW - North Korean Won",
    "KRW - South Korean Won",
    "KWD - Kuwaiti Dinar",
    "KYD - Cayman Islands Dollar",
    "KZT - Kazakhstani Tenge",
    "LAK - Laotian Kip",
    "LBP - Lebanese Pound",
    "LKR - Sri Lankan Rupee",
    "LRD - Liberian Dollar",
    "LSL - Lesotho Loti",
    "LYD - Libyan Dinar",
    "MAD - Moroccan Dirham",
    "MDL - Moldovan Leu",
    "MGA - Malagasy Ariary",
    "MKD - Macedonian Denar",
    "MMK - Myanmar Kyat",
    "MNT - Mongolian Tögrög",
    "MOP - Macanese Pataca",
    "MRU - Mauritanian Ouguiya",
    "MUR - Mauritian Rupee",
    "MVR - Maldivian Rufiyaa",
    "MWK - Malawian Kwacha",
    "MXN - Mexican Peso",
    "MYR - Malaysian Ringgit",
    "MZN - Mozambican Metical",
    "NAD - Namibian Dollar",
    "NGN - Nigerian Naira",
    "NIO - Nicaraguan Córdoba",
    "NOK - Norwegian Krone",
    "NPR - Nepalese Rupee",
    "NZD - New Zealand Dollar",
    "OMR - Omani Rial",
    "PAB - Panamanian Balboa",
    "PEN - Peruvian Sol",
    "PGK - Papua New Guinean Kina",
    "PHP - Philippine Peso",
    "PKR - Pakistani Rupee",
    "PLN - Polish Zloty",
    "PYG - Paraguayan Guarani",
    "QAR - Qatari Rial",
    "RON - Romanian Leu",
    "RSD - Serbian Dinar",
    "RUB - Russian Ruble",
    "RWF - Rwandan Franc",
    "SAR - Saudi Riyal",
    "SBD - Solomon Islands Dollar",
    "SCR - Seychellois Rupee",
    "SDG - Sudanese Pound",
    "SEK - Swedish Krona",
    "SGD - Singapore Dollar",
    "SHP - Saint Helenian Pound",
    "SLL - Sierra Leonean Leone",
    "SOS - Somali Shilling",
    "SRD - Surinamese Dollar",
    "SSP - South Sudanese Pound",
    "STN - São Tomé and Príncipe Dobra",
    "SYP - Syrian Pound",
    "SZL - Swazi Lilangeni",
    "THB - Thai Baht",
    "TJS - Tajikistani Somoni",
    "TMT - Turkmenistani Manat",
    "TND - Tunisian Dinar",
    "TOP - Tongan Paʻanga",
    "TRY - Turkish Lira",
    "TTD - Trinidad and Tobago Dollar",
    "TWD - New Taiwan Dollar",
    "TZS - Tanzanian Shilling",
    "UAH - Ukrainian Hryvnia",
    "UGX - Ugandan Shilling",
    "USD - United States Dollar",
    "UYU - Uruguayan Peso",
    "UZS - Uzbekistani Som",
    "VES - Venezuelan Bolívar",
    "VND - Vietnamese Dong",
    "VUV - Vanuatu Vatu",
    "WST - Samoan Tala",
    "XAF - Central African CFA Franc",
    "XAG - Silver Ounce",
    "XAU - Gold Ounce",
    "XCD - East Caribbean Dollar",
    "XDR - Special Drawing Rights",
    "XOF - West African CFA Franc",
    "XPF - CFP Franc",
    "YER - Yemeni Rial",
    "ZAR - South African Rand",
    "ZMW - Zambian Kwacha",
    "ZWL - Zimbabwean Dollar"
};

// Example exchange rates for each currency (these rates are illustrative)
private double[] rates = {
    3.67, 87.0, 108.0, 385.0, 1.79, 610.0, 143.0, 1.3, 1.8, 1.7,
    1.8, 2.0, 110.0, 1.8, 0.38, 1981.0, 1.0, 1.5, 6.5, 5.2,
    1.0, 82.0, 1.0, 0.25, 0.65, 1.0, 1.6, 0.91, 1.0, 4.0,
    4056.0, 6.9, 1.0, 0.0, 1.0, 4.2, 0.27, 0.23, 1.0, 1.0,
    0.0, 1.0, 0.3, 0.34, 1.0, 35.0, 1.0, 0.5, 0.15, 1.0,
    1.0, 1.0, 0.4, 0.4, 0.9, 0.9, 1.0, 1.0, 0.2, 0.2,
        1.0, 3.5, 2.0, 3.5, 0.12, 1.0, 0.27, 1.0, 1.0, 0.98,
        0.12, 0.15, 0.11, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0,
        1.0, 1.0, 1.0, 1.0, 1.0, 0.75, 1.0, 1.0, 1.0, 1.0,
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_currency_converter);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        inputAmount = findViewById(R.id.inputAmount);
        fromCurrency = findViewById(R.id.fromCurrency);
        toCurrency = findViewById(R.id.toCurrency);
        convertButton = findViewById(R.id.convertButton);
        resultText = findViewById(R.id.resultText);

        ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_spinner_item,
currencies);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        fromCurrency.setAdapter(adapter);
        toCurrency.setAdapter(adapter);

        convertButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                convertCurrency();
            }
        });
    }

    private void convertCurrency() {
        String input = inputAmount.getText().toString();
        if (input.isEmpty()) {
            resultText.setText("Please enter an amount.");
            return;
        }

        double amount = Double.parseDouble(input);
        int fromPosition = fromCurrency.getSelectedItemPosition();
        int toPosition = toCurrency.getSelectedItemPosition();
        double convertedAmount = amount * (rates[toPosition] / rates[fromPosition]);
        resultText.setText(String.format("Converted Amount: %.2f %s", convertedAmount,
currencies[toPosition]));
    }
}
```

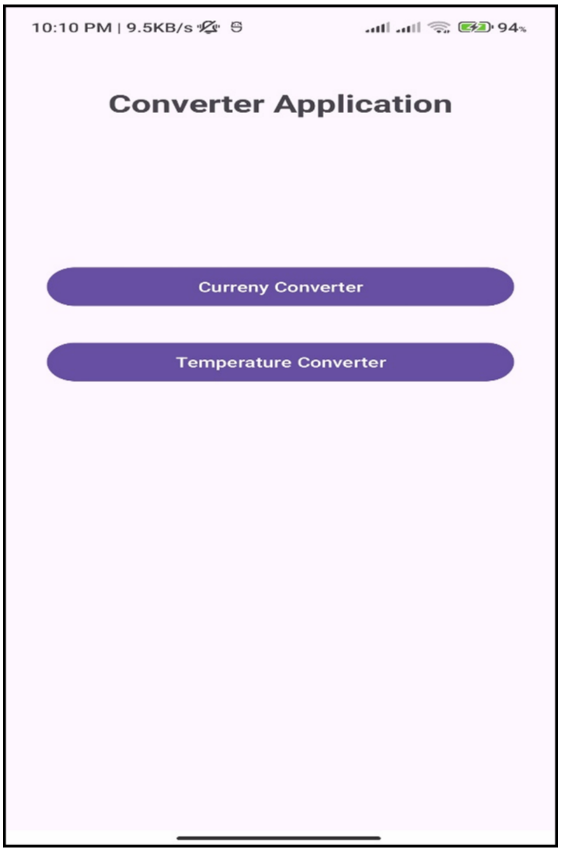
MainActivity.java

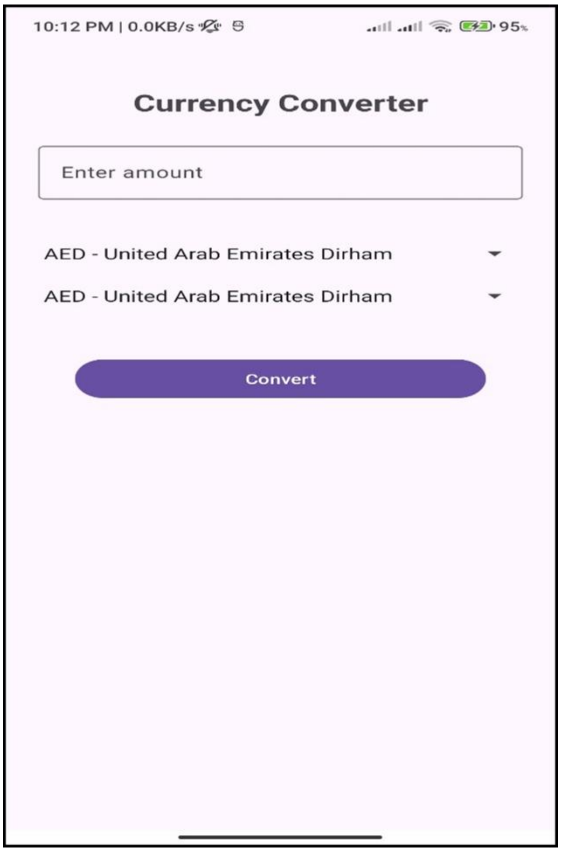
```
package com.example.myapplication;

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

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

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

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        Button currencyConverterButton = findViewById(R.id.button1);
        Button temperatureConverterButton = findViewById(R.id.button2);

        currencyConverterButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, CurrencyConverterActivity.class);
                startActivity(intent);
            }
        });

        temperatureConverterButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, TemperatureConverterActivity.class);
                startActivity(intent);
        }
      });
    }
}
```

TemperatureConverterActivity.java

```
package com.example.myapplication;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import com.google.android.material.textfield.TextInputEditText;
import com.google.android.material.textfield.TextInputLayout;

public class TemperatureConverterActivity extends AppCompatActivity {

    private TextInputEditText celsiusInput;
    private TextInputEditText fahrenheitInput;
    private TextView fahrenheitResult;
    private TextView celsiusResult;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_temperature_converter);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        // Initialize views
        celsiusInput = findViewById(R.id.celsiusInput);
        fahrenheitInput = findViewById(R.id.fahrenheitInput);
        fahrenheitResult = findViewById(R.id.fahrenheitResult);
        celsiusResult = findViewById(R.id.celsiusResult);

        // Set up button listeners
        Button convertToFahrenheit = findViewById(R.id.convertToFahrenheit);
        convertToFahrenheit.setOnClickListener(this::convertCelsiusToFahrenheit);
        Button convertToCelsius = findViewById(R.id.convertToCelsius);
        convertToCelsius.setOnClickListener(this::convertFahrenheitToCelsius);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }

    private void convertCelsiusToFahrenheit(View view) {
        String celsiusStr = celsiusInput.getText().toString();
        if (!celsiusStr.isEmpty()) {
            double celsius = Double.parseDouble(celsiusStr);
            double fahrenheit = (celsius * 9 / 5) + 32;
            fahrenheitResult.setText("Fahrenheit: " + fahrenheit);
        } else {
            fahrenheitResult.setText("Fahrenheit: ");
        }
    }

    private void convertFahrenheitToCelsius(View view) {
        String fahrenheitStr = fahrenheitInput.getText().toString();
        if (!fahrenheitStr.isEmpty()) {
            double fahrenheit = Double.parseDouble(fahrenheitStr);
            double celsius = (fahrenheit - 32) * 5 / 9;
            celsiusResult.setText("Celsius: " + celsius);
        } else {
            celsiusResult.setText("Celsius: ");
        }
    }
}
```

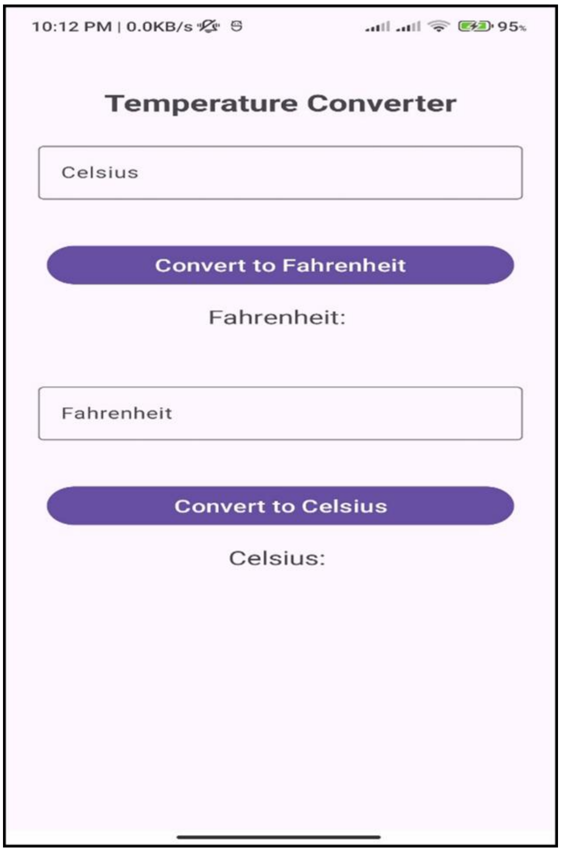
OUTPUT:





AED - United Arab Emirates Dirham

AED - United Arab Emirates Dirham



Practical 7

AIM: Create a Quiz Application with Multiple-Choice Questions. You can Add Features like Score Tracking and Different Categories.

AstronomyQuizActivity.java

```
package com.example.quiz;

import android.app.AlertDialog;
import android.os.Bundle;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class AstronomyQuizActivity extends AppCompatActivity {

   private TextView questionTextView, questionIndexTextView;
   private RadioGroup optionsGroup;
   private Button submitButton;
   private List<AstronomyQuestion> questions;
   private int currentQuestionIndex = 0;
   private int score = 0;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      EdgeToEdge.enable(this);
      setContentView(R.layout.activity_astronomy_quiz);
      ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
         Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
         v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
         return insets;
      });


      questionIndexTextView = findViewById(R.id.questionIndexTextView);
      questionTextView = findViewById(R.id.questionTextView);
```

```
        optionsGroup = findViewById(R.id.optionsGroup);
        submitButton = findViewById(R.id.submitButton);

        generateQuestions();
        displayCurrentQuestion();

        submitButton.setOnClickListener(v -> {
            // Check if an option is selected
            if (optionsGroup.getCheckedRadioButtonId() == -1) {
                // No option is selected, show a Toast error message
                Toast.makeText(this, "Please select an answer before submitting.", Toast.LENGTH_SHORT).show();
            } else {
                // An option is selected, proceed with score calculation
                RadioButton selectedOption = findViewById(optionsGroup.getCheckedRadioButtonId());
                if (selectedOption != null) {
                    if (selectedOption.getText().equals(questions.get(currentQuestionIndex).getCorrectAnswer())) {
                        score++;
                    }
                    currentQuestionIndex++;
                    if (currentQuestionIndex < questions.size()) {
                        displayCurrentQuestion();
                    } else {
                        showScoreDialog();
                    }
                }
            }
        });
    }

    private void generateQuestions() {
        questions = new ArrayList<>();

        // Grade 1-2 Questions (Basic Astronomy)
        questions.add(new AstronomyQuestion("What is the name of our planet?", "Earth", "Mars", "Jupiter",
"Venus", "Earth"));
        questions.add(new AstronomyQuestion("What do we call the star at the center of our solar system?",
"Moon", "Earth", "Sun", "Mars", "Sun"));
        questions.add(new AstronomyQuestion("What do we call the path that the Earth takes around the
Sun?", "Orbit", "Rotation", "Revolution", "Cycle", "Orbit"));
        questions.add(new AstronomyQuestion("What are the big bright lights in the night sky?", "Stars",
"Planets", "Comets", "Galaxies", "Stars"));
        questions.add(new AstronomyQuestion("What do we call the natural satellite that orbits Earth?",
"Star", "Planet", "Moon", "Asteroid", "Moon"));

        // Grade 3-4 Questions (Intermediate Astronomy)
        questions.add(new AstronomyQuestion("What is the largest planet in our solar system?", "Earth",
"Jupiter", "Mars", "Saturn", "Jupiter"));
        questions.add(new AstronomyQuestion("What is the red planet called?", "Earth", "Venus", "Mars",
"Jupiter", "Mars"));
        questions.add(new AstronomyQuestion("What do we call a shooting star?", "Meteor", "Comet",
"Asteroid", "Planet", "Meteor"));
    questions.add(new AstronomyQuestion("What is the name of our galaxy?", "Andromeda", "Milky Way",
"Triangulum", "Whirlpool", "Milky Way"));
    questions.add(new AstronomyQuestion("What is the name of the phase when the moon is fully
illuminated?", "New Moon", "Waxing", "Full Moon", "Waning", "Full Moon"));

    // Grade 5-6 Questions (Advanced Astronomy)
    questions.add(new AstronomyQuestion("What is the name of the closest star to Earth?", "Proxima
Centauri", "Sirius", "Alpha Centauri", "Betelgeuse", "Proxima Centauri"));
    questions.add(new AstronomyQuestion("What force keeps the planets in orbit around the Sun?",
"Friction", "Gravity", "Magnetism", "Inertia", "Gravity"));
    questions.add(new AstronomyQuestion("What do we call a large group of stars, gas, and dust held
together by gravity?", "Star", "Planet", "Galaxy", "Nebula", "Galaxy"));
    questions.add(new AstronomyQuestion("What is the process called when a star forms?", "Supernova",
"Stellar Evolution", "Black Hole", "Nebula", "Stellar Evolution"));
    questions.add(new AstronomyQuestion("What is the name of the telescope that orbits Earth?",
"Hubble", "Kepler", "Chandra", "Spitzer", "Hubble"));

    // Grade 7-8 Questions (Astrophysics)
    questions.add(new AstronomyQuestion("What do we call the theory that explains the origin of the
universe?", "Big Bang Theory", "Steady State Theory", "Cosmic Inflation", "Quantum Theory", "Big Bang
Theory"));
    questions.add(new AstronomyQuestion("What are the remnants of massive stars that can collapse into
a black hole?", "Neutron Stars", "White Dwarfs", "Black Holes", "Red Giants", "Black Holes"));
    questions.add(new AstronomyQuestion("What is the name of the visible part of the Sun?",
"Photosphere", "Chromosphere", "Corona", "Core", "Photosphere"));
    questions.add(new AstronomyQuestion("What is a supernova?", "A dying star explosion", "A type of
black hole", "A comet", "A planet", "A dying star explosion"));
    questions.add(new AstronomyQuestion("What is the Kuiper Belt?", "A region of space beyond
Neptune", "A planet", "A comet", "A star cluster", "A region of space beyond Neptune"));

    // Grade 9-10 Questions (Advanced Astrophysics)
    questions.add(new AstronomyQuestion("What is a black hole?", "A star that has collapsed", "A type of
galaxy", "A planet", "A comet", "A star that has collapsed"));
    questions.add(new AstronomyQuestion("What do we call the theory that describes the expansion of
the universe?", "Cosmic Inflation", "Big Bang Theory", "Quantum Theory", "Relativity", "Big Bang Theory"));
    questions.add(new AstronomyQuestion("What phenomenon is caused by the Earth's shadow on the
Moon?", "Solar Eclipse", "Lunar Eclipse", "Planetary Transit", "Meteor Shower", "Lunar Eclipse"));
    questions.add(new AstronomyQuestion("What is the primary component of the Sun?", "Hydrogen",
"Oxygen", "Helium", "Carbon Dioxide", "Hydrogen"));
    questions.add(new AstronomyQuestion("What do astronomers call a planet that orbits another star?",
"Exoplanet", "Dwarf Planet", "Gas Giant", "Rocky Planet", "Exoplanet"));

    // Grade 11-12 Questions (Theoretical Astronomy)
    questions.add(new AstronomyQuestion("What is dark matter?", "Visible matter in the universe",
"Matter that does not emit light", "Black holes", "Dark stars", "Matter that does not emit light"));
    questions.add(new AstronomyQuestion("What is the name of the theory that describes how galaxies
form?", "Galaxy Evolution Theory", "Big Bang Theory", "Stellar Formation Theory", "Cosmic Microwave
Background", "Galaxy Evolution Theory"));
    questions.add(new AstronomyQuestion("What is a neutron star?", "A star with no gravity", "A
supernova remnant with neutrons", "A black hole", "A type of planet", "A supernova remnant with
neutrons"));
        questions.add(new AstronomyQuestion("What is the cosmic microwave background radiation?",
"Radiation from the Sun", "Afterglow of the Big Bang", "Radiation from black holes", "Radiation from stars",
"Afterglow of the Big Bang"));
        questions.add(new AstronomyQuestion("What is the largest volcano in the solar system?", "Mount
Everest", "Olympus Mons", "Mauna Kea", "Mount Kilimanjaro", "Olympus Mons"));

        // Additional Questions to reach 50 total
        questions.add(new AstronomyQuestion("What do we call a celestial body that orbits a planet?",
"Satellite", "Star", "Comet", "Asteroid", "Satellite"));
        questions.add(new AstronomyQuestion("What is the most abundant element in the universe?",
"Oxygen", "Carbon", "Hydrogen", "Helium", "Hydrogen"));
        questions.add(new AstronomyQuestion("What is the brightest planet visible from Earth?", "Mars",
"Venus", "Jupiter", "Saturn", "Venus"));
        questions.add(new AstronomyQuestion("What do we call the boundary around a black hole beyond
which nothing can escape?", "Event Horizon", "Singularity", "Nebula", "Accretion Disk", "Event Horizon"));
        questions.add(new AstronomyQuestion("What is the main purpose of a space probe?", "To explore
other planets", "To collect meteorites", "To study the Sun", "To observe stars", "To explore other planets"));
        questions.add(new AstronomyQuestion("What is a comet primarily made of?", "Rock and Metal", "Ice
and Dust", "Gas", "Liquid", "Ice and Dust"));
        questions.add(new AstronomyQuestion("What is the approximate age of the universe?", "4.5 billion
years", "10 billion years", "13.8 billion years", "1 billion years", "13.8 billion years"));
        questions.add(new AstronomyQuestion("What is the term for a group of stars forming a pattern?", "Star
Cluster", "Constellation", "Galaxy", "Nebula", "Constellation"));
        questions.add(new AstronomyQuestion("What is the name of the first human-made satellite launched
into space?", "Apollo 11", "Voyager 1", "Sputnik 1", "Hubble", "Sputnik 1"));
        questions.add(new AstronomyQuestion("What is the study of the universe beyond Earth called?",
"Geology", "Astronomy", "Astrophysics", "Cosmology", "Astronomy"));

        // Shuffle questions for randomness
        Collections.shuffle(questions);
        questions = questions.subList(0, 10); // Get only 10 random questions
    }

    private void displayCurrentQuestion() {
        AstronomyQuestion currentQuestion = questions.get(currentQuestionIndex);
        questionTextView.setText(currentQuestion.getQuestion());
        questionIndexTextView.setText("Question " + (currentQuestionIndex + 1) + " :");
        optionsGroup.clearCheck();

        // Assuming you have RadioButtons for options
        ((RadioButton) optionsGroup.getChildAt(0)).setText(currentQuestion.getOption1());
        ((RadioButton) optionsGroup.getChildAt(1)).setText(currentQuestion.getOption2());
        ((RadioButton) optionsGroup.getChildAt(2)).setText(currentQuestion.getOption3());
        ((RadioButton) optionsGroup.getChildAt(3)).setText(currentQuestion.getOption4());
    }

    private void showScoreDialog() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Quiz Completed");
        builder.setMessage("Your score is: " + score + " out of " + questions.size());
        builder.setPositiveButton("OK", (dialog, which) -> {
            finish(); // Go back to the previous activity
        });
        builder.setCancelable(false);
        AlertDialog dialog = builder.create();
        dialog.show();
    }
}
```

ComputerQuizActivity.java

```
package com.example.quiz;

import android.app.AlertDialog;
import android.os.Bundle;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class ComputerQuizActivity extends AppCompatActivity {

    private TextView questionTextView, questionIndexTextView;
    private RadioGroup optionsGroup;
    private Button submitButton;
    private List<ComputerQuestion> questions;
    private int currentQuestionIndex = 0;
    private int score = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_computer_quiz);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });


        questionIndexTextView = findViewById(R.id.questionIndexTextView);
        questionTextView = findViewById(R.id.questionTextView);
        optionsGroup = findViewById(R.id.optionsGroup);
        submitButton = findViewById(R.id.submitButton);

        generateQuestions();
        displayCurrentQuestion();

        submitButton.setOnClickListener(v -> {
            // Check if an option is selected
            if (optionsGroup.getCheckedRadioButtonId() == -1) {
                // No option is selected, show a Toast error message
                Toast.makeText(this, "Please select an answer before submitting.", Toast.LENGTH_SHORT).show();
            } else {
                // An option is selected, proceed with score calculation
                RadioButton selectedOption = findViewById(optionsGroup.getCheckedRadioButtonId());
                if (selectedOption != null) {
                    if (selectedOption.getText().equals(questions.get(currentQuestionIndex).getCorrectAnswer())) {
                        score++;
                    }
                    currentQuestionIndex++;
                    if (currentQuestionIndex < questions.size()) {
                        displayCurrentQuestion();
                    } else {
                        showScoreDialog();
                    }
                }
            }
        });
    }

    private void generateQuestions() {
        questions = new ArrayList<>();

        // Grade 1-2 Questions (Basic Computer Science)
        questions.add(new ComputerQuestion("What is a computer?", "A machine that helps us work", "A type of food", "A pet", "A toy", "A machine that helps us work"));
        questions.add(new ComputerQuestion("What do we use to type on a computer?", "Mouse", "Keyboard", "Screen", "Printer", "Keyboard"));
        questions.add(new ComputerQuestion("Which part of the computer displays images?", "Keyboard", "Monitor", "Mouse", "CPU", "Monitor"));
        questions.add(new ComputerQuestion("What is the main function of a computer mouse?", "To display images", "To move the cursor", "To print documents", "To save files", "To move the cursor"));
        questions.add(new ComputerQuestion("What do you call the programs that run on a computer?", "Hardware", "Software", "Peripherals", "Networks", "Software"));

        // Grade 3-4 Questions (Intermediate Computer Science)
        questions.add(new ComputerQuestion("What does 'Internet' refer to?", "A type of software", "A global network of computers", "A hardware component", "A programming language", "A global network of computers"));
        questions.add(new ComputerQuestion("What is the function of an antivirus program?", "To create documents", "To protect against viruses", "To increase speed", "To browse the internet", "To protect against viruses"));
        questions.add(new ComputerQuestion("Which one is an example of an operating system?", "Google Chrome", "Windows", "Microsoft Word", "Adobe Photoshop", "Windows"));
        questions.add(new ComputerQuestion("What does 'URL' stand for?", "Uniform Resource Locator", "Universal Resource Locator", "Unique Resource Link", "Universal Reference Link", "Uniform Resource Locator"));
        questions.add(new ComputerQuestion("What does a web browser do?", "Stores files", "Browses the internet", "Processes data", "Displays images", "Browses the internet"));

        // Grade 5-6 Questions (Advanced Computer Science)
        questions.add(new ComputerQuestion("What is coding?", "Writing instructions for a computer", "Making music", "Designing graphics", "Playing games", "Writing instructions for a computer"));
        questions.add(new ComputerQuestion("Which language is commonly used for web development?", "Python", "HTML", "Java", "C++", "HTML"));
        questions.add(new ComputerQuestion("What is a 'bug' in programming?", "A feature", "An error", "A tool", "A type of software", "An error"));
        questions.add(new ComputerQuestion("What does 'RAM' stand for?", "Read Access Memory", "Random Access Memory", "Rapid Access Memory", "Readily Accessible Memory", "Random Access Memory"));
        questions.add(new ComputerQuestion("What is the function of a compiler?", "To convert code into executable programs", "To debug code", "To store files", "To display images", "To convert code into executable programs"));

        // Grade 7-8 Questions (Programming and Algorithms)
        questions.add(new ComputerQuestion("What is an algorithm?", "A recipe for solving problems", "A type of computer", "A software application", "A programming language", "A recipe for solving problems"));
        questions.add(new ComputerQuestion("Which of the following is a loop structure in programming?", "If statement", "For loop", "Variable", "Function", "For loop"));
        questions.add(new ComputerQuestion("What is the purpose of a function in programming?", "To store data", "To perform a specific task", "To display output", "To read input", "To perform a specific task"));
        questions.add(new ComputerQuestion("What does 'HTML' stand for?", "HyperText Markup Language", "HyperText Memory Language", "HighText Markup Language", "Hyper Transfer Markup Language", "HyperText Markup Language"));
        questions.add(new ComputerQuestion("In Python, what symbol is used for comments?", "#", "//", "/*", "<!--", "#"));

        // Grade 9-10 Questions (Advanced Programming Concepts)
        questions.add(new ComputerQuestion("What is the main purpose of a database?", "To store data", "To display images", "To run applications", "To print documents", "To store data"));
        questions.add(new ComputerQuestion("What is object-oriented programming?", "A way to organize code using objects", "A type of programming language", "A coding error", "A software application", "A way to organize code using objects"));
        questions.add(new ComputerQuestion("Which of the following is a markup language?", "Java", "C++", "HTML", "Python", "HTML"));
        questions.add(new ComputerQuestion("What does 'API' stand for?", "Application Programming
Interface", "Advanced Programming Instruction", "Application Protocol Interface", "Automated Program Interface", "Application Programming Interface"));
        questions.add(new ComputerQuestion("In Java, what is used to create a class?", "class keyword", "function keyword", "object keyword", "module keyword", "class keyword"));

        // Grade 11-12 Questions (Advanced Topics)
        questions.add(new ComputerQuestion("What is the purpose of encryption?", "To speed up data processing", "To secure data", "To compress files", "To store data", "To secure data"));
        questions.add(new ComputerQuestion("What is machine learning?", "A way to program computers to learn from data", "A method for writing code", "A type of hardware", "A programming language", "A way to program computers to learn from data"));
        questions.add(new ComputerQuestion("What is the main function of the CPU?", "To store data", "To process instructions", "To display graphics", "To connect to the internet", "To process instructions"));
        questions.add(new ComputerQuestion("Which of the following is a programming paradigm?", "Imperative", "Declarative", "Functional", "All of the above", "All of the above"));
        questions.add(new ComputerQuestion("What is an IP address?", "A unique identifier for a device on a network", "A type of virus", "A programming tool", "A web browser", "A unique identifier for a device on a network"));

        // Additional Questions to reach 50 total
        questions.add(new ComputerQuestion("What is the purpose of a firewall?", "To protect against unauthorized access", "To connect to the internet", "To run applications", "To store data", "To protect against unauthorized access"));
        questions.add(new ComputerQuestion("Which language is primarily used for data analysis?", "Java", "Python", "C++", "HTML", "Python"));
        questions.add(new ComputerQuestion("What does 'IoT' stand for?", "Internet of Things", "Internet of Technology", "Internal Operating Tool", "Integrated Online Technology", "Internet of Things"));
        questions.add(new ComputerQuestion("What is a server?", "A device that provides data to other devices", "A type of software", "A computer game", "A network cable", "A device that provides data to other devices"));
        questions.add(new ComputerQuestion("Which of the following is a database management system?", "HTML", "MySQL", "Python", "JavaScript", "MySQL"));
        questions.add(new ComputerQuestion("What is cloud computing?", "Storing data on the internet", "A type of software", "A programming language", "A hardware component", "Storing data on the internet"));
        questions.add(new ComputerQuestion("What does 'SEO' stand for?", "Search Engine Optimization", "Software Engineering Online", "Simple Electronic Output", "Server Enhanced Operation", "Search Engine Optimization"));
        questions.add(new ComputerQuestion("What is a variable in programming?", "A storage location for data", "A type of software", "A coding error", "An input device", "A storage location for data"));
        questions.add(new ComputerQuestion("What is debugging?", "Finding and fixing errors in code", "Running a program", "Writing code", "Storing data", "Finding and fixing errors in code"));
        questions.add(new ComputerQuestion("What is a network?", "A group of interconnected devices", "A type of software", "A programming language", "A storage device", "A group of interconnected devices"));

        // Shuffle questions for randomness
        Collections.shuffle(questions);
        questions = questions.subList(0, 10); // Get only 10 random questions
    }

    private void displayCurrentQuestion() {
        ComputerQuestion currentQuestion = questions.get(currentQuestionIndex);
        questionTextView.setText(currentQuestion.getQuestion());
        questionIndexTextView.setText("Question " + (currentQuestionIndex + 1) + " :");
        optionsGroup.clearCheck();

        // Assuming you have RadioButtons for options
        ((RadioButton) optionsGroup.getChildAt(0)).setText(currentQuestion.getOption1());
        ((RadioButton) optionsGroup.getChildAt(1)).setText(currentQuestion.getOption2());
        ((RadioButton) optionsGroup.getChildAt(2)).setText(currentQuestion.getOption3());
        ((RadioButton) optionsGroup.getChildAt(3)).setText(currentQuestion.getOption4());
    }

    private void showScoreDialog() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Quiz Completed");
        builder.setMessage("Your score is: " + score + " out of " + questions.size());
        builder.setPositiveButton("OK", (dialog, which) -> {
            finish(); // Go back to the previous activity
        });
        builder.setCancelable(false);
        AlertDialog dialog = builder.create();
        dialog.show();
    }
}
```

DatabaseHelper.java

```
package com.example.quiz;

import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DatabaseHelper extends SQLiteOpenHelper {

    private static final String DATABASE_NAME = "quiz.db";
    private static final int DATABASE_VERSION = 1;
    private static final String TABLE_NAME = "scores";
    private static final String COLUMN_ID = "id";
    private static final String COLUMN_SCORE = "score";
    private static final String COLUMN_CATEGORY = "category";

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTable = "CREATE TABLE " + TABLE_NAME + " (" +
                COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
                COLUMN_SCORE + " INTEGER, " +
                COLUMN_CATEGORY + " TEXT)";
        db.execSQL(createTable);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }

    public void insertScore(int score, String category) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(COLUMN_SCORE, score);
        values.put(COLUMN_CATEGORY, category);
        db.insert(TABLE_NAME, null, values);
        db.close();
    }
}
```

GeographyQuizActivity.java

```
package com.example.quiz;

import android.app.AlertDialog;
import android.os.Bundle;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class GeographyQuizActivity extends AppCompatActivity {

    private TextView questionTextView, questionIndexTextView;
    private RadioGroup optionsGroup;
    private Button submitButton;
    private List<GeographyQuestion> questions;
    private int currentQuestionIndex = 0;
    private int score = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_geography_quiz);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });


        questionIndexTextView = findViewById(R.id.questionIndexTextView);
        questionTextView = findViewById(R.id.questionTextView);
        optionsGroup = findViewById(R.id.optionsGroup);
        submitButton = findViewById(R.id.submitButton);

        generateQuestions();
        displayCurrentQuestion();

        submitButton.setOnClickListener(v -> {
            // Check if an option is selected
            if (optionsGroup.getCheckedRadioButtonId() == -1) {
                // No option is selected, show a Toast error message
                Toast.makeText(this, "Please select an answer before submitting.", Toast.LENGTH_SHORT).show();
            } else {
                // An option is selected, proceed with score calculation
                RadioButton selectedOption = findViewById(optionsGroup.getCheckedRadioButtonId());
                if (selectedOption != null) {
                    if (selectedOption.getText().equals(questions.get(currentQuestionIndex).getCorrectAnswer())) {
                        score++;
                    }
                    currentQuestionIndex++;
                    if (currentQuestionIndex < questions.size()) {
                        displayCurrentQuestion();
                    } else {
                        showScoreDialog();
                    }
                }
            }
        });
    }

    private void generateQuestions() {
        questions = new ArrayList<>();

        // Grade 1-2 Questions (Basic Geography)
        questions.add(new GeographyQuestion("What is the name of the planet we live on?", "Mars", "Earth",
"Venus", "Jupiter", "Earth"));
```

```
    questions.add(new GeographyQuestion("What ocean is the largest in the world?", "Indian Ocean",
"Atlantic Ocean", "Arctic Ocean", "Pacific Ocean", "Pacific Ocean"));
    questions.add(new GeographyQuestion("Which continent is known as the 'Dark Continent'?", "Asia",
"Africa", "Europe", "Australia", "Africa"));
    questions.add(new GeographyQuestion("What is the capital city of France?", "London", "Berlin",
"Madrid", "Paris", "Paris"));
    questions.add(new GeographyQuestion("What type of map shows physical features?", "Political Map",
"Physical Map", "Weather Map", "Topographic Map", "Physical Map"));

    // Grade 3-4 Questions (Intermediate Geography)
    questions.add(new GeographyQuestion("Which country is known as the Land of the Rising Sun?",
"China", "Japan", "South Korea", "Thailand", "Japan"));
    questions.add(new GeographyQuestion("What is the longest river in the world?", "Amazon River", "Nile
River", "Yangtze River", "Mississippi River", "Nile River"));
    questions.add(new GeographyQuestion("Which desert is the largest in the world?", "Sahara Desert",
"Gobi Desert", "Kalahari Desert", "Atacama Desert", "Sahara Desert"));
    questions.add(new GeographyQuestion("Which country has the most natural lakes?", "Canada",
"United States", "Russia", "India", "Canada"));
    questions.add(new GeographyQuestion("What is the capital of Australia?", "Sydney", "Canberra",
"Melbourne", "Brisbane", "Canberra"));

    // Grade 5-6 Questions (Advanced Geography)
    questions.add(new GeographyQuestion("Which continent is known as the Frozen Continent?", "Asia",
"North America", "Antarctica", "Australia", "Antarctica"));
    questions.add(new GeographyQuestion("What is the main language spoken in Brazil?", "Spanish",
"Portuguese", "English", "French", "Portuguese"));
    questions.add(new GeographyQuestion("What mountain range separates Europe and Asia?", "Andes",
"Rocky Mountains", "Himalayas", "Ural Mountains", "Ural Mountains"));
    questions.add(new GeographyQuestion("Which river runs through Egypt?", "Nile", "Amazon",
"Yangtze", "Mississippi", "Nile"));
    questions.add(new GeographyQuestion("What is the capital of Italy?", "Rome", "Venice", "Milan",
"Florence", "Rome"));

    // Grade 7-8 Questions (Geography Concepts)
    questions.add(new GeographyQuestion("What is the process by which water vapor turns into liquid
water?", "Evaporation", "Condensation", "Precipitation", "Transpiration", "Condensation"));
    questions.add(new GeographyQuestion("Which country has the largest population?", "India", "United
States", "China", "Indonesia", "China"));
    questions.add(new GeographyQuestion("What is the term for a large body of salt water surrounded by
land?", "Lake", "Ocean", "Sea", "Bay", "Sea"));
    questions.add(new GeographyQuestion("What is the capital city of Canada?", "Toronto", "Ottawa",
"Vancouver", "Montreal", "Ottawa"));
    questions.add(new GeographyQuestion("Which country is both a continent and a country?",
"Australia", "Antarctica", "Africa", "Greenland", "Australia"));

    // Grade 9-10 Questions (Human Geography)
    questions.add(new GeographyQuestion("What is urbanization?", "Movement of people to rural areas",
"Increase in population in cities", "Decrease in population", "Movement of animals", "Increase in population
in cities"));
    questions.add(new GeographyQuestion("Which of the following is a factor of push migration?", "Job
opportunities", "Natural disasters", "Climate stability", "Political stability", "Natural disasters"));
    questions.add(new GeographyQuestion("What is the term for the boundaries set by countries?", "Geopolitics", "Geography", "Geographical Boundaries", "Political Borders", "Political Borders"));
    questions.add(new GeographyQuestion("What is the main reason for deforestation?", "Urbanization", "Agriculture", "Mining", "All of the above", "All of the above"));
    questions.add(new GeographyQuestion("Which economic system is based on private ownership?", "Capitalism", "Socialism", "Communism", "Feudalism", "Capitalism"));

    // Grade 11-12 Questions (Advanced Geography)
    questions.add(new GeographyQuestion("What is globalization?", "Isolation of countries", "Increase in international trade", "Decrease in population", "All of the above", "Increase in international trade"));
    questions.add(new GeographyQuestion("Which country is the largest by land area?", "United States", "Canada", "China", "Russia", "Russia"));
    questions.add(new GeographyQuestion("What type of map shows the distribution of various features?", "Thematic Map", "Topographic Map", "Political Map", "Physical Map", "Thematic Map"));
    questions.add(new GeographyQuestion("What is the primary cause of climate change?", "Deforestation", "Industrialization", "Burning fossil fuels", "All of the above", "All of the above"));
    questions.add(new GeographyQuestion("Which geographical term refers to the flat area near a river?", "Delta", "Plateau", "Plain", "Mountain", "Delta"));

    // Additional Questions to reach 50 total
    questions.add(new GeographyQuestion("What do we call the imaginary line that divides the Earth into Northern and Southern Hemispheres?", "Equator", "Prime Meridian", "Tropic of Cancer", "Tropic of Capricorn", "Equator"));
    questions.add(new GeographyQuestion("What is the term for the study of the Earth's surface and the processes that shape it?", "Geology", "Geography", "Geophysics", "Meteorology", "Geography"));
    questions.add(new GeographyQuestion("Which is the highest mountain in the world?", "K2", "Kangchenjunga", "Mount Everest", "Lhotse", "Mount Everest"));
    questions.add(new GeographyQuestion("Which of the following is a landlocked country?", "Austria", "Brazil", "Egypt", "Italy", "Austria"));
    questions.add(new GeographyQuestion("What is a biome?", "A large naturally occurring community of flora and fauna", "An area with similar climate", "A city with a high population", "Both A and B", "Both A and B"));
    questions.add(new GeographyQuestion("Which region is known for having the most biodiversity?", "Deserts", "Tropical Rainforests", "Tundra", "Grasslands", "Tropical Rainforests"));
    questions.add(new GeographyQuestion("What is the capital of Egypt?", "Cairo", "Alexandria", "Giza", "Luxor", "Cairo"));
    questions.add(new GeographyQuestion("What continent is the Sahara Desert located on?", "Asia", "North America", "Africa", "Australia", "Africa"));
    questions.add(new GeographyQuestion("Which city is known as the Big Apple?", "Los Angeles", "Chicago", "New York City", "Miami", "New York City"));
    questions.add(new GeographyQuestion("What do we call the line that measures 0 degrees longitude?", "Prime Meridian", "Equator", "Tropic of Cancer", "International Date Line", "Prime Meridian"));

    // Shuffle questions for randomness
    Collections.shuffle(questions);
    questions = questions.subList(0, 10); // Get only 10 random questions
}

private void displayCurrentQuestion() {
    GeographyQuestion currentQuestion = questions.get(currentQuestionIndex);
    questionTextView.setText(currentQuestion.getQuestion());
    questionIndexTextView.setText("Question " + (currentQuestionIndex + 1) + " :");
    optionsGroup.clearCheck();

    // Assuming you have RadioButtons for options
    ((RadioButton) optionsGroup.getChildAt(0)).setText(currentQuestion.getOption1());
    ((RadioButton) optionsGroup.getChildAt(1)).setText(currentQuestion.getOption2());
    ((RadioButton) optionsGroup.getChildAt(2)).setText(currentQuestion.getOption3());
    ((RadioButton) optionsGroup.getChildAt(3)).setText(currentQuestion.getOption4());
}

private void showScoreDialog() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Quiz Completed");
    builder.setMessage("Your score is: " + score + " out of " + questions.size());
    builder.setPositiveButton("OK", (dialog, which) -> {
        finish(); // Go back to the previous activity
    });
    builder.setCancelable(false);
    AlertDialog dialog = builder.create();
    dialog.show();
}
}
```

HistoryQuizActivity.java

```
package com.example.quiz;

import android.app.AlertDialog;
import android.os.Bundle;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class HistoryQuizActivity extends AppCompatActivity {

    private TextView questionTextView, questionIndexTextView;
    private RadioGroup optionsGroup;
    private Button submitButton;
    private List<HistoryQuestion> questions;
    private int currentQuestionIndex = 0;
    private int score = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_history_quiz);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });


        questionIndexTextView = findViewById(R.id.questionIndexTextView);
        questionTextView = findViewById(R.id.questionTextView);
        optionsGroup = findViewById(R.id.optionsGroup);
        submitButton = findViewById(R.id.submitButton);

        generateQuestions();
        displayCurrentQuestion();

        submitButton.setOnClickListener(v -> {
            // Check if an option is selected
            if (optionsGroup.getCheckedRadioButtonId() == -1) {
                // No option is selected, show a Toast error message
                Toast.makeText(this, "Please select an answer before submitting.", Toast.LENGTH_SHORT).show();
            } else {
                // An option is selected, proceed with score calculation
                RadioButton selectedOption = findViewById(optionsGroup.getCheckedRadioButtonId());
                if (selectedOption != null) {
                    if (selectedOption.getText().equals(questions.get(currentQuestionIndex).getCorrectAnswer())) {
                        score++;
                    }
                    currentQuestionIndex++;
                    if (currentQuestionIndex < questions.size()) {
                        displayCurrentQuestion();
                    } else {
                        showScoreDialog();
                    }
                }
            }
        });
    }

    private void generateQuestions() {
        questions = new ArrayList<>();
        // Grade 1-2 Questions (Basic History)
        questions.add(new HistoryQuestion("Who was the first President of the United States?", "George Washington", "Thomas Jefferson", "Abraham Lincoln", "John Adams", "George Washington"));
        questions.add(new HistoryQuestion("What is the capital of the United States?", "New York", "Washington, D.C.", "Los Angeles", "Chicago", "Washington, D.C."));
        questions.add(new HistoryQuestion("Which continent is known as the 'Dark Continent'?", "Asia", "Africa", "Europe", "Australia", "Africa"));
        questions.add(new HistoryQuestion("Who was the first man on the moon?", "Buzz Aldrin", "Yuri Gagarin", "Neil Armstrong", "John Glenn", "Neil Armstrong"));
        questions.add(new HistoryQuestion("What was the name of the ship that carried the Pilgrims to America?", "The Mayflower", "The Nina", "The Pinta", "The Santa Maria", "The Mayflower"));

        // Grade 3-4 Questions (Intermediate History)
        questions.add(new HistoryQuestion("What was the main reason for the American Revolution?", "Taxation without representation", "Desire for independence", "Religious freedom", "Land disputes", "Taxation without representation"));
        questions.add(new HistoryQuestion("Who wrote the Declaration of Independence?", "George Washington", "Benjamin Franklin", "Thomas Jefferson", "John Adams", "Thomas Jefferson"));
        questions.add(new HistoryQuestion("Which war was fought to end slavery in the United States?", "World War I", "World War II", "The Civil War", "The Revolutionary War", "The Civil War"));
        questions.add(new HistoryQuestion("What year did World War I begin?", "1914", "1918", "1939", "1945", "1914"));
        questions.add(new HistoryQuestion("What was the name of the first permanent English settlement in America?", "Jamestown", "Plymouth", "Roanoke", "Boston", "Jamestown"));

        // Grade 5-6 Questions (Advanced History)
        questions.add(new HistoryQuestion("Who was the first female Prime Minister of the United Kingdom?", "Indira Gandhi", "Margaret Thatcher", "Golda Meir", "Angela Merkel", "Margaret Thatcher"));
        questions.add(new HistoryQuestion("What was the main cause of the Cold War?", "Nuclear arms race", "Political ideologies", "Territorial disputes", "Economic competition", "Political ideologies"));
        questions.add(new HistoryQuestion("What significant event happened on July 20, 1969?", "End of World War II", "First man on the moon", "Fall of the Berlin Wall", "Assassination of JFK", "First man on the moon"));
        questions.add(new HistoryQuestion("Who was known as the 'Father of History'?", "Herodotus", "Pliny the Elder", "Thucydides", "Tacitus", "Herodotus"));
        questions.add(new HistoryQuestion("Which empire was ruled by Genghis Khan?", "Roman Empire", "Ottoman Empire", "Mongol Empire", "Byzantine Empire", "Mongol Empire"));

        // Grade 7-8 Questions (Modern History)
        questions.add(new HistoryQuestion("Who was the leader of the Soviet Union during World War II?", "Stalin", "Trotsky", "Lenin", "Gorbachev", "Stalin"));
        questions.add(new HistoryQuestion("What was the primary reason for the fall of the Berlin Wall?", "Economic decline", "Political pressure", "Civil unrest", "Reunification", "Political pressure"));
        questions.add(new HistoryQuestion("Which event marked the start of the Great Depression?", "Stock Market Crash of 1929", "World War I", "World War II", "Dust Bowl", "Stock Market Crash of 1929"));
        questions.add(new HistoryQuestion("What was the main goal of the civil rights movement?", "End segregation", "Women's rights", "Voting rights", "Labor rights", "End segregation"));
        questions.add(new HistoryQuestion("Who was the first African American president of the United States?", "Bill Clinton", "George W. Bush", "Barack Obama", "Jimmy Carter", "Barack Obama"));
        // Grade 9-10 Questions (Contemporary History)
        questions.add(new HistoryQuestion("What was the main outcome of the Treaty of Versailles?", "End of
    World War I", "Start of World War II", "Formation of the United Nations", "Cold War beginnings", "End of
    World War I"));
        questions.add(new HistoryQuestion("What was the primary cause of the Vietnam War?",
    "Communism", "Imperialism", "Nationalism", "Terrorism", "Communism"));
        questions.add(new HistoryQuestion("What major event took place on September 11, 2001?", "Fall of
    the Berlin Wall", "Terrorist attacks on the U.S.", "End of the Cold War", "First Gulf War", "Terrorist attacks
    on the U.S."));
        questions.add(new HistoryQuestion("Who was Nelson Mandela?", "A U.S. president", "A South African
    anti-apartheid revolutionary", "A British Prime Minister", "An Indian independence leader", "A South African
    anti-apartheid revolutionary"));
        questions.add(new HistoryQuestion("Which country was the first to grant women the right to vote?",
    "United States", "New Zealand", "Finland", "Australia", "New Zealand"));

        // Grade 11-12 Questions (Advanced History)
        questions.add(new HistoryQuestion("What was the main focus of the Enlightenment?", "Reason and
    individualism", "Tradition", "Religion", "Warfare", "Reason and individualism"));
        questions.add(new HistoryQuestion("Which historical figure is known for the quote, 'Give me liberty, or
    give me death'?", "Patrick Henry", "George Washington", "Thomas Jefferson", "John Adams", "Patrick
    Henry"));
        questions.add(new HistoryQuestion("What was the primary cause of World War II?", "Treaty of
    Versailles", "Nationalism", "Imperialism", "Failure of the League of Nations", "Treaty of Versailles"));
        questions.add(new HistoryQuestion("Who was the main author of the Communist Manifesto?", "Karl
    Marx", "Friedrich Engels", "Leon Trotsky", "Vladimir Lenin", "Karl Marx"));
        questions.add(new HistoryQuestion("What major event triggered the start of World War I?",
    "Assassination of Archduke Franz Ferdinand", "German invasion of Poland", "Bombing of Pearl Harbor", "The
    sinking of the Lusitania", "Assassination of Archduke Franz Ferdinand"));

        // Additional Questions to reach 50 total
        questions.add(new HistoryQuestion("What year did the United States declare independence?", "1776",
    "1783", "1775", "1787", "1776"));
        questions.add(new HistoryQuestion("Who was the first woman to fly solo across the Atlantic Ocean?",
    "Amelia Earhart", "Bessie Coleman", "Harriet Quimby", "Eleanor Roosevelt", "Amelia Earhart"));
        questions.add(new HistoryQuestion("What is the significance of the Magna Carta?", "First document to
    limit the power of the king", "First document for women's rights", "First constitution", "First bill of rights",
    "First document to limit the power of the king"));
        questions.add(new HistoryQuestion("Who was the leader of the Nazi Party?", "Adolf Hitler", "Hermann
    Göring", "Joseph Goebbels", "Heinrich Himmler", "Adolf Hitler"));
        questions.add(new HistoryQuestion("What was the primary goal of the Marshall Plan?", "Rebuild
    Europe after World War II", "End the Great Depression", "Provide aid to Asia", "Promote Communism",
    "Rebuild Europe after World War II"));
        questions.add(new HistoryQuestion("What is the name of the British passenger ship that was sunk by a
    German U-boat in 1915?", "Titanic", "Lusitania", "Queen Mary", "Britannic", "Lusitania"));
        questions.add(new HistoryQuestion("Who was the first man to circumnavigate the globe?", "Ferdinand
    Magellan", "Christopher Columbus", "James Cook", "Francis Drake", "Ferdinand Magellan"));
        questions.add(new HistoryQuestion("What event led to the start of the American Civil War?", "Election
    of Abraham Lincoln", "Fugitive Slave Act", "Attack on Fort Sumter", "Dred Scott decision", "Attack on Fort
    Sumter"));
```

```
        questions.add(new HistoryQuestion("What is the main focus of the United Nations?", "Global peace and security", "Trade agreements", "Cultural exchange", "Environmental issues", "Global peace and security"));
        questions.add(new HistoryQuestion("What was the main outcome of the Spanish-American War?", "U.S. expansion overseas", "Independence for Cuba", "End of colonialism", "Spanish territorial gains", "U.S. expansion overseas"));

        // Shuffle questions for randomness
        Collections.shuffle(questions);
        questions = questions.subList(0, 10); // Get only 10 random questions
    }

    private void displayCurrentQuestion() {
        HistoryQuestion currentQuestion = questions.get(currentQuestionIndex);
        questionTextView.setText(currentQuestion.getQuestion());
        questionIndexTextView.setText("Question " + (currentQuestionIndex + 1) + " :");
        optionsGroup.clearCheck();

        // Assuming you have RadioButtons for options
        ((RadioButton) optionsGroup.getChildAt(0)).setText(currentQuestion.getOption1());
        ((RadioButton) optionsGroup.getChildAt(1)).setText(currentQuestion.getOption2());
        ((RadioButton) optionsGroup.getChildAt(2)).setText(currentQuestion.getOption3());
        ((RadioButton) optionsGroup.getChildAt(3)).setText(currentQuestion.getOption4());
    }

    private void showScoreDialog() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Quiz Completed");
        builder.setMessage("Your score is: " + score + " out of " + questions.size());
        builder.setPositiveButton("OK", (dialog, which) -> {
            finish(); // Go back to the previous activity
        });
        builder.setCancelable(false);
        AlertDialog dialog = builder.create();
        dialog.show();
    }
}
```

LiteratureQuizActivity.java

```
package com.example.quiz;

import android.app.AlertDialog;
import android.os.Bundle;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class LiteratureQuizActivity extends AppCompatActivity {

    private TextView questionTextView, questionIndexTextView;
    private RadioGroup optionsGroup;
    private Button submitButton;
    private List<LiteratureQuestion> questions;
    private int currentQuestionIndex = 0;
    private int score = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_literature_quiz);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });


        questionIndexTextView = findViewById(R.id.questionIndexTextView);
        questionTextView = findViewById(R.id.questionTextView);
        optionsGroup = findViewById(R.id.optionsGroup);
        submitButton = findViewById(R.id.submitButton);

        generateQuestions();
        displayCurrentQuestion();

        submitButton.setOnClickListener(v -> {
            // Check if an option is selected
            if (optionsGroup.getCheckedRadioButtonId() == -1) {
                // No option is selected, show a Toast error message
                Toast.makeText(this, "Please select an answer before submitting.", Toast.LENGTH_SHORT).show();
            } else {
                // An option is selected, proceed with score calculation
                RadioButton selectedOption = findViewById(optionsGroup.getCheckedRadioButtonId());
                if (selectedOption != null) {
                    if (selectedOption.getText().equals(questions.get(currentQuestionIndex).getCorrectAnswer())) {
                        score++;
                    }
                    currentQuestionIndex++;
```

```
          if (currentQuestionIndex < questions.size()) {
            displayCurrentQuestion();
          } else {
            showScoreDialog();
          }
        }
      }
    });
  }

  private void generateQuestions() {
    questions = new ArrayList<>();

    // Grade 1-2 Questions (Basic Literature)
    questions.add(new LiteratureQuestion("What is the title of the famous bear who loves honey?",
"Winnie the Pooh", "Paddington Bear", "Yogi Bear", "Baloo", "Winnie the Pooh"));
    questions.add(new LiteratureQuestion("Who wrote 'The Cat in the Hat'?", "Dr. Seuss", "A.A. Milne",
"J.K. Rowling", "Roald Dahl", "Dr. Seuss"));
    questions.add(new LiteratureQuestion("What color is the Big Bird in 'Sesame Street'?", "Red", "Yellow",
"Blue", "Green", "Yellow"));
    questions.add(new LiteratureQuestion("Which character is known for saying 'To infinity and beyond'?",
"Woody", "Buzz Lightyear", "Shrek", "Simba", "Buzz Lightyear"));
    questions.add(new LiteratureQuestion("What kind of animal is 'Charlotte' in 'Charlotte's Web'?", "Dog",
"Pig", "Spider", "Mouse", "Spider"));

    // Grade 3-4 Questions (Intermediate Literature)
    questions.add(new LiteratureQuestion("Who is the author of 'Harry Potter'?", "J.R.R. Tolkien", "J.K.
Rowling", "C.S. Lewis", "Stephen King", "J.K. Rowling"));
    questions.add(new LiteratureQuestion("In 'The Lion, the Witch and the Wardrobe', who is the main
villain?", "Aslan", "Edmund", "The White Witch", "Peter", "The White Witch"));
    questions.add(new LiteratureQuestion("What is the name of the place where Peter Pan lives?",
"Neverland", "Wonderland", "Oz", "Hogwarts", "Neverland"));
    questions.add(new LiteratureQuestion("Which fairy tale features a girl who loses her shoe?",
"Cinderella", "Snow White", "Sleeping Beauty", "Little Red Riding Hood", "Cinderella"));
    questions.add(new LiteratureQuestion("Who wrote 'Charlie and the Chocolate Factory'?", "Roald Dahl",
"J.K. Rowling", "Dr. Seuss", "Lewis Carroll", "Roald Dahl"));

    // Grade 5-6 Questions (Advanced Literature)
    questions.add(new LiteratureQuestion("In 'To Kill a Mockingbird', who is the narrator?", "Scout Finch",
"Atticus Finch", "Jem Finch", "Boo Radley", "Scout Finch"));
    questions.add(new LiteratureQuestion("What is the main theme of 'The Great Gatsby'?", "Friendship",
"Love", "The American Dream", "Family", "The American Dream"));
    questions.add(new LiteratureQuestion("Which book begins with the line 'Call me Ishmael'?", "Moby
Dick", "The Old Man and the Sea", "Huckleberry Finn", "1984", "Moby Dick"));
    questions.add(new LiteratureQuestion("What type of animal is 'Animal Farm' primarily about?", "Dogs",
"Cats", "Pigs", "Horses", "Pigs"));
    questions.add(new LiteratureQuestion("Who wrote 'The Chronicles of Narnia' series?", "J.K. Rowling",
"C.S. Lewis", "J.R.R. Tolkien", "Philip Pullman", "C.S. Lewis"));

    // Grade 7-8 Questions (Literary Analysis and Genres)
```

```
        questions.add(new LiteratureQuestion("What genre does 'The Hobbit' belong to?", "Fiction", "Fantasy", "Mystery", "Science Fiction", "Fantasy"));
        questions.add(new LiteratureQuestion("Who is the protagonist in '1984'?", "Winston Smith", "O'Brien", "Julia", "Big Brother", "Winston Smith"));
        questions.add(new LiteratureQuestion("What is the setting of 'Romeo and Juliet'?", "Verona", "London", "Paris", "Rome", "Verona"));
        questions.add(new LiteratureQuestion("What type of poem tells a story?", "Haiku", "Limerick", "Ballad", "Sonnet", "Ballad"));
        questions.add(new LiteratureQuestion("In 'The Catcher in the Rye', what is Holden Caulfield searching for?", "Happiness", "Truth", "Identity", "Adventure", "Identity"));

        // Grade 9-10 Questions (Classic Literature)
        questions.add(new LiteratureQuestion("Who wrote 'Pride and Prejudice'?", "Charlotte Brontë", "Jane Austen", "Emily Brontë", "Mary Shelley", "Jane Austen"));
        questions.add(new LiteratureQuestion("In 'Moby Dick', what is Captain Ahab obsessed with?", "Treasure", "Whales", "The sea", "Revenge", "Whales"));
        questions.add(new LiteratureQuestion("Which novel features the character Jay Gatsby?", "The Great Gatsby", "The Grapes of Wrath", "The Old Man and the Sea", "The Catcher in the Rye", "The Great Gatsby"));
        questions.add(new LiteratureQuestion("What is the main message of 'Animal Farm'?", "Power corrupts", "Friendship is important", "Education is key", "Love conquers all", "Power corrupts"));
        questions.add(new LiteratureQuestion("What is the main conflict in 'Romeo and Juliet'?", "Man vs. Society", "Man vs. Man", "Man vs. Nature", "Man vs. Self", "Man vs. Society"));

        // Grade 11-12 Questions (Advanced Literary Concepts)
        questions.add(new LiteratureQuestion("What is the central idea of a literary work called?", "Theme", "Plot", "Character", "Setting", "Theme"));
        questions.add(new LiteratureQuestion("In literature, what does 'imagery' refer to?", "Visual representation", "Sound devices", "Metaphors", "Character development", "Visual representation"));
        questions.add(new LiteratureQuestion("Who wrote 'Frankenstein'?", "Mary Shelley", "Charlotte Brontë", "Jane Austen", "Emily Dickinson", "Mary Shelley"));
        questions.add(new LiteratureQuestion("What is a sonnet?", "A type of novel", "A narrative poem", "A 14-line poem", "A long epic", "A 14-line poem"));
        questions.add(new LiteratureQuestion("What is a recurring element in a literary work called?", "Theme", "Motive", "Symbol", "Motif", "Motif"));

        // Additional Questions to reach 50 total
        questions.add(new LiteratureQuestion("What type of story is 'The Odyssey'?", "Epic", "Tragedy", "Comedy", "Drama", "Epic"));
        questions.add(new LiteratureQuestion("Which character says, 'To be, or not to be'?", "Macbeth", "Hamlet", "Othello", "Romeo", "Hamlet"));
        questions.add(new LiteratureQuestion("What is the term for a character's struggle against an outside force?", "Internal conflict", "External conflict", "Climax", "Resolution", "External conflict"));
        questions.add(new LiteratureQuestion("Who wrote 'The Bell Jar'?", "Sylvia Plath", "Virginia Woolf", "Emily Dickinson", "Zora Neale Hurston", "Sylvia Plath"));
        questions.add(new LiteratureQuestion("In which novel do characters attend a mysterious party in West Egg?", "The Great Gatsby", "The Catcher in the Rye", "Brave New World", "The Old Man and the Sea", "The Great Gatsby"));
        questions.add(new LiteratureQuestion("What does 'stream of consciousness' refer to in literature?", "A plot device", "A character's thoughts", "A writing style", "A narrative technique", "A writing style"));
        questions.add(new LiteratureQuestion("Which novel features the character Hester Prynne?", "The
    Scarlet Letter", "The Great Gatsby", "Moby Dick", "Pride and Prejudice", "The Scarlet Letter"));
        questions.add(new LiteratureQuestion("Who is known for the phrase 'I have a dream'?", "Malcolm X",
    "Martin Luther King Jr.", "Nelson Mandela", "Abraham Lincoln", "Martin Luther King Jr."));
        questions.add(new LiteratureQuestion("What does the term 'bildungsroman' mean?", "A coming-of-
    age story", "A tragedy", "A comedy", "An epic", "A coming-of-age story"));
        questions.add(new LiteratureQuestion("In Shakespeare's plays, what is the term for a character's
    speech alone on stage?", "Soliloquy", "Monologue", "Dialogue", "Aside", "Soliloquy"));

        // Shuffle questions for randomness
        Collections.shuffle(questions);
        questions = questions.subList(0, 10); // Get only 10 random questions
      }

      private void displayCurrentQuestion() {
        LiteratureQuestion currentQuestion = questions.get(currentQuestionIndex);
        questionTextView.setText(currentQuestion.getQuestion());
        questionIndexTextView.setText("Question " + (currentQuestionIndex + 1) + " :");
        optionsGroup.clearCheck();

        // Assuming you have RadioButtons for options
        ((RadioButton) optionsGroup.getChildAt(0)).setText(currentQuestion.getOption1());
        ((RadioButton) optionsGroup.getChildAt(1)).setText(currentQuestion.getOption2());
        ((RadioButton) optionsGroup.getChildAt(2)).setText(currentQuestion.getOption3());
        ((RadioButton) optionsGroup.getChildAt(3)).setText(currentQuestion.getOption4());
      }

      private void showScoreDialog() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Quiz Completed");
        builder.setMessage("Your score is: " + score + " out of " + questions.size());
        builder.setPositiveButton("OK", (dialog, which) -> {
          finish(); // Go back to the previous activity
        });
        builder.setCancelable(false);
        AlertDialog dialog = builder.create();
        dialog.show();
      }
    }
```

MainActivity.java

```
package com.example.quiz;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

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

    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
      Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
      v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
      return insets;
    });

    Button buttonMaths = findViewById(R.id.buttonMaths);
    Button buttonScience = findViewById(R.id.buttonScience);
    Button buttonHistory = findViewById(R.id.buttonHistory);
    Button buttonGeography = findViewById(R.id.buttonGeography);
    Button buttonLiterature = findViewById(R.id.buttonLiterature);
    Button buttonAstronomy = findViewById(R.id.buttonAstronomy);
    Button buttonSports = findViewById(R.id.buttonSports);
    Button buttonComputer = findViewById(R.id.buttonComputer);


    buttonMaths.setOnClickListener(v -> {
      Intent intent = new Intent(MainActivity.this, MathsQuizActivity.class);
      startActivity(intent);
    });

    buttonScience.setOnClickListener(v -> {
      Intent intent = new Intent(MainActivity.this, ScienceQuizActivity.class);
      startActivity(intent);
    });

    buttonHistory.setOnClickListener(v -> {
      Intent intent = new Intent(MainActivity.this, HistoryQuizActivity.class);
      startActivity(intent);
    });

    buttonGeography.setOnClickListener(v -> {
      Intent intent = new Intent(MainActivity.this, GeographyQuizActivity.class);
      startActivity(intent);
    });

    buttonLiterature.setOnClickListener(v -> {
      Intent intent = new Intent(MainActivity.this, LiteratureQuizActivity.class);
      startActivity(intent);
    });
```

```
        buttonAstronomy.setOnClickListener(v -> {
            Intent intent = new Intent(MainActivity.this, AstronomyQuizActivity.class);
            startActivity(intent);
        });

        buttonSports.setOnClickListener(v -> {
            Intent intent = new Intent(MainActivity.this, SportsQuizActivity.class);
            startActivity(intent);
        });

        buttonComputer.setOnClickListener(v -> {
            Intent intent = new Intent(MainActivity.this, ComputerQuizActivity.class);
            startActivity(intent);
        });
    }
}
```

MathsQuizActivity.java

```
package com.example.quiz;

import android.app.AlertDialog;
import android.os.Bundle;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class MathsQuizActivity extends AppCompatActivity {

    private TextView questionTextView, questionIndexTextView;
    private RadioGroup optionsGroup;
    private Button submitButton;
    private List<MathsQuestion> questions;
    private int currentQuestionIndex = 0;
    private int score = 0;
    private String category = "Math"; // Specify your category here

    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maths_quiz);
    EdgeToEdge.enable(this);

    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.maths_quiz), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });

    questionIndexTextView = findViewById(R.id.questionIndexTextView);
    questionTextView = findViewById(R.id.questionTextView);
    optionsGroup = findViewById(R.id.optionsGroup);
    submitButton = findViewById(R.id.submitButton);

    generateQuestions();
    displayCurrentQuestion();

    submitButton.setOnClickListener(v -> {
        // Check if an option is selected
        if (optionsGroup.getCheckedRadioButtonId() == -1) {
            // No option is selected, show a Toast error message
            Toast.makeText(this, "Please select an answer before submitting.", Toast.LENGTH_SHORT).show();
        } else {
            // An option is selected, proceed with score calculation
            RadioButton selectedOption = findViewById(optionsGroup.getCheckedRadioButtonId());
            if (selectedOption != null) {
                if (selectedOption.getText().equals(questions.get(currentQuestionIndex).getCorrectAnswer())) {
                    score++;
                }
                currentQuestionIndex++;
                if (currentQuestionIndex < questions.size()) {
                    displayCurrentQuestion();
                } else {
                    showScoreDialog();
                }
            }
        }
    });
}

private void generateQuestions() {
    questions = new ArrayList<>();

    // Grade 1-2 Questions (Basic Arithmetic)
    questions.add(new MathsQuestion("What is 2 + 3?", "5", "6", "4", "3", "5"));
    questions.add(new MathsQuestion("What is 8 - 2?", "6", "5", "4", "3", "6"));
    questions.add(new MathsQuestion("What is 3 * 4?", "12", "10", "9", "15", "12"));
    questions.add(new MathsQuestion("What is 10 / 2?", "5", "6", "4", "3", "5"));
```

```
        questions.add(new MathsQuestion("What is 7 + 1?", "8", "7", "6", "9", "8"));

        // Grade 3-4 Questions (Intermediate Arithmetic)
        questions.add(new MathsQuestion("What is 9 - 5?", "4", "3", "2", "5", "4"));
        questions.add(new MathsQuestion("What is 6 * 6?", "36", "30", "32", "34", "36"));
        questions.add(new MathsQuestion("What is 24 / 6?", "4", "5", "6", "3", "4"));
        questions.add(new MathsQuestion("What is 15 + 7?", "22", "20", "18", "19", "22"));
        questions.add(new MathsQuestion("What is 36 - 14?", "22", "21", "20", "19", "22"));

        // Grade 5-6 Questions (Fractions and Decimals)
        questions.add(new MathsQuestion("What is 1/2 + 1/4?", "3/4", "1/2", "1/4", "1", "3/4"));
        questions.add(new MathsQuestion("What is 0.5 + 0.75?", "1.25", "1.5", "1", "1.75", "1.25"));
        questions.add(new MathsQuestion("What is 3/5 - 1/5?", "2/5", "1/5", "3/5", "4/5", "2/5"));
        questions.add(new MathsQuestion("What is 2.5 * 2?", "5", "4", "6", "3", "5"));
        questions.add(new MathsQuestion("What is 12.8 - 4.3?", "8.5", "7.5", "9", "8", "8.5"));

        // Grade 7-8 Questions (Algebra and Ratios)
        questions.add(new MathsQuestion("What is 3x + 2 = 11? What is x?", "3", "4", "5", "2", "3"));
        questions.add(new MathsQuestion("What is the ratio of 8 to 4?", "2:1", "1:2", "4:1", "1:4", "2:1"));
        questions.add(new MathsQuestion("If a = 5, what is 2a + 3?", "13", "10", "8", "5", "13"));
        questions.add(new MathsQuestion("What is 5% of 200?", "10", "5", "15", "20", "10"));
        questions.add(new MathsQuestion("Solve for y: 2y - 4 = 10.", "7", "5", "8", "6", "7"));

        // Grade 9-10 Questions (Geometry and Advanced Algebra)
        questions.add(new MathsQuestion("What is the area of a triangle with base 10 and height 5?", "25",
"30", "20", "15", "25"));
        questions.add(new MathsQuestion("What is the circumference of a circle with radius 3?", "18.84",
"9.42", "12.56", "6.28", "18.84"));
        questions.add(new MathsQuestion("Solve for x: x^2 - 4 = 0.", "2", "4", "0", "8", "2"));
        questions.add(new MathsQuestion("What is the volume of a cube with side length 3?", "27", "9", "18",
"12", "27"));
        questions.add(new MathsQuestion("What is the slope of the line passing through (2, 3) and (4, 7)?", "2",
"1", "3", "4", "2"));

        // Grade 11-12 Questions (Calculus and Trigonometry)
        questions.add(new MathsQuestion("What is the derivative of x^2?", "2x", "x", "x^2", "3x", "2x"));
        questions.add(new MathsQuestion("What is the integral of 3x?", "1.5x^2 + C", "3x^2 + C", "x^3 + C",
"0.5x^2 + C", "1.5x^2 + C"));
        questions.add(new MathsQuestion("What is sin(90 degrees)?", "1", "0", "0.5", "undefined", "1"));
        questions.add(new MathsQuestion("What is the cosine of 45 degrees?", "0.707", "1", "0.5", "0.866",
"0.707"));
        questions.add(new MathsQuestion("What is the limit of (x^2 - 1)/(x - 1) as x approaches 1?", "2", "1",
"0", "undefined", "2"));

        // Add more variety to reach 50 questions
        questions.add(new MathsQuestion("What is the product of 8 and 9?", "72", "70", "64", "80", "72"));
        questions.add(new MathsQuestion("If x = 4, what is 3x + 2?", "14", "12", "10", "16", "14"));
        questions.add(new MathsQuestion("What is the hypotenuse of a right triangle with legs 3 and 4?", "5",
"6", "7", "8", "5"));
        questions.add(new MathsQuestion("Solve for x: 5x - 7 = 18.", "5", "6", "4", "3", "5"));
        questions.add(new MathsQuestion("What is the square root of 144?", "12", "10", "14", "16", "12"));
        questions.add(new MathsQuestion("What is the area of a rectangle with length 4 and width 5?", "20",
"18", "25", "15", "20"));
        questions.add(new MathsQuestion("What is 2^3?", "8", "6", "7", "9", "8"));
        questions.add(new MathsQuestion("What is the sum of angles in a triangle?", "180", "90", "360", "270",
"180"));
        questions.add(new MathsQuestion("What is the greatest common factor of 24 and 36?", "12", "6", "18",
"3", "12"));
        questions.add(new MathsQuestion("What is 50% of 80?", "40", "20", "60", "30", "40"));

        // Shuffle questions for randomness
        Collections.shuffle(questions);
        questions = questions.subList(0, 10); // Get only 10 random questions
    }

    private void displayCurrentQuestion() {
        MathsQuestion currentQuestion = questions.get(currentQuestionIndex);
        questionTextView.setText(currentQuestion.getQuestion());
        questionIndexTextView.setText("Question " + (currentQuestionIndex + 1) + " :");
        optionsGroup.clearCheck();

        // Assuming you have RadioButtons for options
        ((RadioButton) optionsGroup.getChildAt(0)).setText(currentQuestion.getOption1());
        ((RadioButton) optionsGroup.getChildAt(1)).setText(currentQuestion.getOption2());
        ((RadioButton) optionsGroup.getChildAt(2)).setText(currentQuestion.getOption3());
        ((RadioButton) optionsGroup.getChildAt(3)).setText(currentQuestion.getOption4());
    }

    private void showScoreDialog() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Quiz Completed");
        builder.setMessage("Your score is: " + score + " out of " + questions.size());
        builder.setPositiveButton("OK", (dialog, which) -> {
            saveScoreToDatabase(score, category); // Pass the category when saving the score
            finish(); // Go back to the previous activity
        });
        builder.setCancelable(false);
        AlertDialog dialog = builder.create();
        dialog.show();
    }

    private void saveScoreToDatabase(int score, String category) {
        DatabaseHelper dbHelper = new DatabaseHelper(this);
        dbHelper.insertScore(score, category);
    }
}
```

ScienceQuizActivity.java

```
package com.example.quiz;
```

```
import android.app.AlertDialog;
import android.os.Bundle;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class ScienceQuizActivity extends AppCompatActivity {

    private TextView questionTextView, questionIndexTextView;
    private RadioGroup optionsGroup;
    private Button submitButton;
    private List<ScienceQuestion> questions;
    private int currentQuestionIndex = 0;
    private int score = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_science_quiz);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        questionIndexTextView = findViewById(R.id.questionIndexTextView);
        questionTextView = findViewById(R.id.questionTextView);
        optionsGroup = findViewById(R.id.optionsGroup);
        submitButton = findViewById(R.id.submitButton);

        generateQuestions();
        displayCurrentQuestion();

        submitButton.setOnClickListener(v -> {
            // Check if an option is selected
            if (optionsGroup.getCheckedRadioButtonId() == -1) {
                // No option is selected, show a Toast error message
                Toast.makeText(this, "Please select an answer before submitting.", Toast.LENGTH_SHORT).show();
            } else {
                // An option is selected, proceed with score calculation
                RadioButton selectedOption = findViewById(optionsGroup.getCheckedRadioButtonId());
                if (selectedOption != null) {
                    if (selectedOption.getText().equals(questions.get(currentQuestionIndex).getCorrectAnswer())) {
                        score++;
                    }
                    currentQuestionIndex++;
                    if (currentQuestionIndex < questions.size()) {
                        displayCurrentQuestion();
                    } else {
                        showScoreDialog();
                    }
                }
            }
        });
    }

    private void generateQuestions() {
        questions = new ArrayList<>();

        // Grade 1-2 Questions (Basic Science)
        questions.add(new ScienceQuestion("What do plants need to grow?", "Sunlight", "Water", "Soil", "All
of the above", "All of the above"));
        questions.add(new ScienceQuestion("What is the main gas that we breathe?", "Oxygen", "Hydrogen",
"Carbon Dioxide", "Nitrogen", "Oxygen"));
        questions.add(new ScienceQuestion("What is the largest planet in our solar system?", "Earth", "Mars",
"Jupiter", "Venus", "Jupiter"));
        questions.add(new ScienceQuestion("What is the force that pulls objects towards the Earth?",
"Magnetism", "Friction", "Gravity", "Inertia", "Gravity"));
        questions.add(new ScienceQuestion("What do bees produce?", "Honey", "Wax", "Nectar", "Pollen",
"Honey"));

        // Grade 3-4 Questions (Intermediate Science)
        questions.add(new ScienceQuestion("What is the process by which plants make their food?",
"Photosynthesis", "Respiration", "Digestion", "Evaporation", "Photosynthesis"));
        questions.add(new ScienceQuestion("What is the chemical symbol for water?", "H2O", "O2", "CO2",
"H2", "H2O"));
        questions.add(new ScienceQuestion("What is the primary source of energy for the Earth?", "The
Moon", "The Sun", "The Wind", "The Earth itself", "The Sun"));
        questions.add(new ScienceQuestion("What do we call animals that eat both plants and meat?",
"Herbivores", "Carnivores", "Omnivores", "Detritivores", "Omnivores"));
        questions.add(new ScienceQuestion("What is the process of changing from a liquid to a gas?",
"Condensation", "Evaporation", "Sublimation", "Precipitation", "Evaporation"));

        // Grade 5-6 Questions (Advanced Science)
        questions.add(new ScienceQuestion("What is the main component of the Earth's atmosphere?",
"Oxygen", "Nitrogen", "Carbon Dioxide", "Helium", "Nitrogen"));
        questions.add(new ScienceQuestion("What do we call the movement of the Earth around the Sun?",
"Rotation", "Revolution", "Translation", "Orbiting", "Revolution"));
    questions.add(new ScienceQuestion("What type of energy is stored in food?", "Kinetic Energy",
"Nuclear Energy", "Chemical Energy", "Solar Energy", "Chemical Energy"));
    questions.add(new ScienceQuestion("What is the powerhouse of the cell?", "Nucleus", "Mitochondria",
"Ribosome", "Chloroplast", "Mitochondria"));
    questions.add(new ScienceQuestion("What are the building blocks of matter?", "Atoms", "Molecules",
"Cells", "Elements", "Atoms"));

    // Grade 7-8 Questions (Biology and Chemistry)
    questions.add(new ScienceQuestion("What is the pH level of pure water?", "7", "5", "9", "10", "7"));
    questions.add(new ScienceQuestion("What is the smallest unit of life?", "Atom", "Cell", "Molecule",
"Tissue", "Cell"));
    questions.add(new ScienceQuestion("What is the chemical formula for carbon dioxide?", "CO2", "C2O",
"CO", "O2C", "CO2"));
    questions.add(new ScienceQuestion("What is the term for the transfer of heat through direct contact?",
"Conduction", "Convection", "Radiation", "Insulation", "Conduction"));
    questions.add(new ScienceQuestion("What part of the plant conducts photosynthesis?", "Roots",
"Stems", "Leaves", "Flowers", "Leaves"));

    // Grade 9-10 Questions (Physics and Chemistry)
    questions.add(new ScienceQuestion("What is Newton's second law of motion?", "F=ma", "E=mc^2",
"pV=nRT", "v=at", "F=ma"));
    questions.add(new ScienceQuestion("What do we call a change in the genetic structure of an
organism?", "Mutation", "Evolution", "Adaptation", "Hybridization", "Mutation"));
    questions.add(new ScienceQuestion("What is the chemical symbol for gold?", "Au", "Ag", "Pb", "Fe",
"Au"));
    questions.add(new ScienceQuestion("What is the process by which plants release water vapor into the
air?", "Transpiration", "Respiration", "Photosynthesis", "Condensation", "Transpiration"));
    questions.add(new ScienceQuestion("What is the main function of red blood cells?", "Transport
oxygen", "Fight infection", "Clot blood", "Transport nutrients", "Transport oxygen"));

    // Grade 11-12 Questions (Advanced Biology and Physics)
    questions.add(new ScienceQuestion("What is the primary function of DNA?", "Energy storage",
"Genetic information storage", "Cellular respiration", "Protein synthesis", "Genetic information storage"));
    questions.add(new ScienceQuestion("What is the main idea behind natural selection?", "Survival of the
fittest", "Adaptation", "Competition", "Mutation", "Survival of the fittest"));
    questions.add(new ScienceQuestion("What is the term for the energy of motion?", "Potential Energy",
"Kinetic Energy", "Thermal Energy", "Nuclear Energy", "Kinetic Energy"));
    questions.add(new ScienceQuestion("What do we call the process of converting light energy into
chemical energy in plants?", "Photosynthesis", "Respiration", "Fermentation", "Combustion",
"Photosynthesis"));
    questions.add(new ScienceQuestion("What is the main gas responsible for climate change?", "Oxygen",
"Carbon Dioxide", "Nitrogen", "Methane", "Carbon Dioxide"));

    // Additional Questions to reach 50 total
    questions.add(new ScienceQuestion("What is the primary function of the immune system?", "To digest
food", "To protect against disease", "To circulate blood", "To transport oxygen", "To protect against
disease"));
    questions.add(new ScienceQuestion("What is the unit of force?", "Joule", "Newton", "Watt", "Pascal",
"Newton"));
```

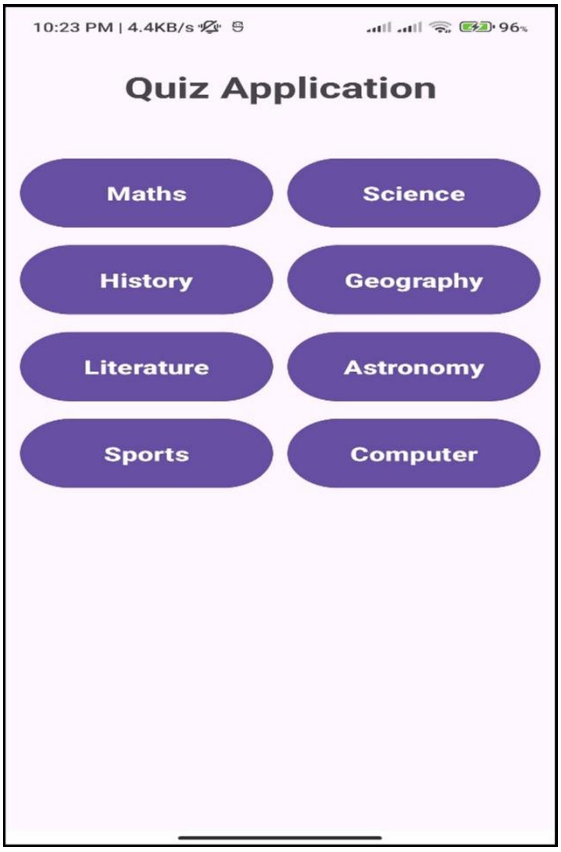
```
        questions.add(new ScienceQuestion("What is the force that opposes motion between two surfaces in
contact?", "Friction", "Gravity", "Momentum", "Acceleration", "Friction"));
        questions.add(new ScienceQuestion("What is the scientific study of heredity called?", "Genetics",
"Evolution", "Ecology", "Anatomy", "Genetics"));
        questions.add(new ScienceQuestion("What is the primary role of enzymes in the body?", "Speed up
chemical reactions", "Break down food", "Provide energy", "Store nutrients", "Speed up chemical
reactions"));
        questions.add(new ScienceQuestion("What is the primary cause of tides on Earth?", "The Moon", "The
Sun", "Wind", "Earth's rotation", "The Moon"));
        questions.add(new ScienceQuestion("What is the most abundant gas in the Earth's atmosphere?",
"Oxygen", "Carbon Dioxide", "Nitrogen", "Argon", "Nitrogen"));
        questions.add(new ScienceQuestion("What is the name of the process by which cells obtain energy
from glucose?", "Photosynthesis", "Cellular Respiration", "Fermentation", "Digestion", "Cellular
Respiration"));
        questions.add(new ScienceQuestion("What is the largest organ in the human body?", "Heart", "Liver",
"Skin", "Lungs", "Skin"));
        questions.add(new ScienceQuestion("What is the basic structural and functional unit of life?", "Cell",
"Tissue", "Organ", "System", "Cell"));

        // Shuffle questions for randomness
        Collections.shuffle(questions);
        questions = questions.subList(0, 10); // Get only 10 random questions
    }

    private void displayCurrentQuestion() {
        ScienceQuestion currentQuestion = questions.get(currentQuestionIndex);
        questionTextView.setText(currentQuestion.getQuestion());
        questionIndexTextView.setText("Question " + (currentQuestionIndex + 1) + " :");
        optionsGroup.clearCheck();

        // Assuming you have RadioButtons for options
        ((RadioButton) optionsGroup.getChildAt(0)).setText(currentQuestion.getOption1());
        ((RadioButton) optionsGroup.getChildAt(1)).setText(currentQuestion.getOption2());
        ((RadioButton) optionsGroup.getChildAt(2)).setText(currentQuestion.getOption3());
        ((RadioButton) optionsGroup.getChildAt(3)).setText(currentQuestion.getOption4());
    }

    private void showScoreDialog() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Quiz Completed");
        builder.setMessage("Your score is: " + score + " out of " + questions.size());
        builder.setPositiveButton("OK", (dialog, which) -> {
            finish(); // Go back to the previous activity
        });
        builder.setCancelable(false);
        AlertDialog dialog = builder.create();
        dialog.show();
    }
}
```

SportsQuizActivity.java

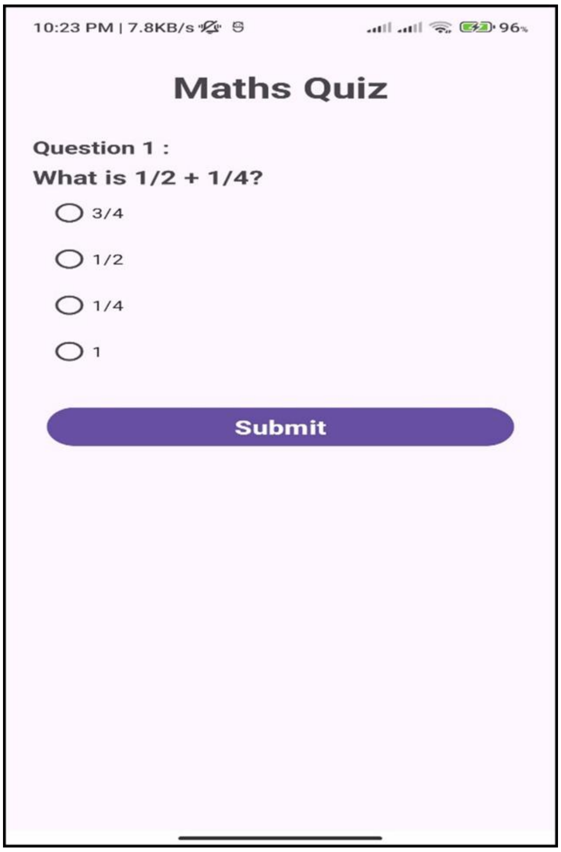
```
package com.example.quiz;

import android.app.AlertDialog;
import android.os.Bundle;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class SportsQuizActivity extends AppCompatActivity {

    private TextView questionTextView, questionIndexTextView;
    private RadioGroup optionsGroup;
    private Button submitButton;
    private List<SportsQuestion> questions;
    private int currentQuestionIndex = 0;
    private int score = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_sports_quiz);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });


        questionIndexTextView = findViewById(R.id.questionIndexTextView);
        questionTextView = findViewById(R.id.questionTextView);
        optionsGroup = findViewById(R.id.optionsGroup);
        submitButton = findViewById(R.id.submitButton);

        generateQuestions();
        displayCurrentQuestion();
    submitButton.setOnClickListener(v -> {
      // Check if an option is selected
      if (optionsGroup.getCheckedRadioButtonId() == -1) {
        // No option is selected, show a Toast error message
        Toast.makeText(this, "Please select an answer before submitting.", Toast.LENGTH_SHORT).show();
      } else {
        // An option is selected, proceed with score calculation
        RadioButton selectedOption = findViewById(optionsGroup.getCheckedRadioButtonId());
        if (selectedOption != null) {
          if (selectedOption.getText().equals(questions.get(currentQuestionIndex).getCorrectAnswer())) {
            score++;
          }
          currentQuestionIndex++;
          if (currentQuestionIndex < questions.size()) {
            displayCurrentQuestion();
          } else {
            showScoreDialog();
          }
        }
      }
    });
  }

  private void generateQuestions() {
    questions = new ArrayList<>();

    // Grade 1-2 Questions (Basic Sports Knowledge)
    questions.add(new SportsQuestion("What is the name of the sport played with a round ball and two
goals?", "Soccer", "Basketball", "Baseball", "Tennis", "Soccer"));
    questions.add(new SportsQuestion("How many players are on a soccer team?", "7", "9", "11", "12",
"11"));
    questions.add(new SportsQuestion("In which sport do players try to hit a ball over a net?", "Basketball",
"Tennis", "Baseball", "Soccer", "Tennis"));
    questions.add(new SportsQuestion("What sport involves running and jumping over hurdles?", "Rugby",
"Track and Field", "Swimming", "Football", "Track and Field"));
    questions.add(new SportsQuestion("What do you call the person who referees a soccer match?",
"Umpire", "Referee", "Judge", "Official", "Referee"));

    // Grade 3-4 Questions (Intermediate Sports Knowledge)
    questions.add(new SportsQuestion("What sport uses a racket to hit a shuttlecock?", "Badminton",
"Soccer", "Baseball", "Hockey", "Badminton"));
    questions.add(new SportsQuestion("What is the main objective in basketball?", "To score goals", "To
score runs", "To score points", "To hit the ball", "To score points"));
    questions.add(new SportsQuestion("In which sport do you score by hitting a ball into a net?", "Golf",
"Tennis", "Baseball", "Basketball", "Tennis"));
    questions.add(new SportsQuestion("What is the term for a home run hit in baseball?", "Touchdown",
"Goal", "Home run", "Strike", "Home run"));
    questions.add(new SportsQuestion("How many minutes are in a standard football game?", "60", "90",
"120", "75", "90"));
```

```
    // Grade 5-6 Questions (Advanced Sports Knowledge)
    questions.add(new SportsQuestion("What is the main governing body of football (soccer)?", "FIFA",
"NCAA", "NHL", "NBA", "FIFA"));
    questions.add(new SportsQuestion("What do we call a game that ends with no winner?", "Draw", "Tie",
"Stalemate", "Win", "Draw"));
    questions.add(new SportsQuestion("In what sport would you find a backboard?", "Baseball",
"Basketball", "Football", "Rugby", "Basketball"));
    questions.add(new SportsQuestion("What is the name of the championship game in American
football?", "World Series", "Super Bowl", "Finals", "Playoffs", "Super Bowl"));
    questions.add(new SportsQuestion("What is the term for a basketball shot made from behind the three-
point line?", "Three-pointer", "Dunk", "Layup", "Free throw", "Three-pointer"));

    // Grade 7-8 Questions (Team Sports)
    questions.add(new SportsQuestion("In which sport is the Stanley Cup awarded?", "Football", "Baseball",
"Ice Hockey", "Basketball", "Ice Hockey"));
    questions.add(new SportsQuestion("What is the maximum number of players allowed on a rugby
team?", "15", "13", "11", "7", "15"));
    questions.add(new SportsQuestion("What is the term for the act of running with the ball in soccer?",
"Dribbling", "Passing", "Shooting", "Kicking", "Dribbling"));
    questions.add(new SportsQuestion("Which country hosted the FIFA World Cup in 2018?", "Germany",
"Brazil", "Russia", "South Africa", "Russia"));
    questions.add(new SportsQuestion("What is the sport that combines running, swimming, and cycling?",
"Triathlon", "Decathlon", "Heptathlon", "Biathlon", "Triathlon"));

    // Grade 9-10 Questions (Competitive Sports)
    questions.add(new SportsQuestion("What is the primary objective in boxing?", "To score points", "To
knock out the opponent", "To win rounds", "To stay on your feet", "To knock out the opponent"));
    questions.add(new SportsQuestion("In which sport would you perform a slam dunk?", "Soccer",
"Baseball", "Basketball", "Tennis", "Basketball"));
    questions.add(new SportsQuestion("What is the term for a score of zero in tennis?", "Love", "Nil",
"Zero", "Point", "Love"));
    questions.add(new SportsQuestion("What do you call the player who throws the ball to start the play
in American football?", "Quarterback", "Center", "Running back", "Wide receiver", "Quarterback"));
    questions.add(new SportsQuestion("What is the name of the Olympic Games held every four years?",
"World Championships", "Pan American Games", "Olympics", "Commonwealth Games", "Olympics"));

    // Grade 11-12 Questions (Advanced Sports Strategy)
    questions.add(new SportsQuestion("What is the term for the area on a basketball court where players
score two points?", "Three-point line", "Paint", "Free-throw line", "Arc", "Paint"));
    questions.add(new SportsQuestion("What is the length of a standard marathon?", "21 kilometers", "42
kilometers", "50 kilometers", "35 kilometers", "42 kilometers"));
    questions.add(new SportsQuestion("Which sport features the term 'offside'?", "Soccer", "Baseball",
"Basketball", "Volleyball", "Soccer"));
    questions.add(new SportsQuestion("What is the main objective of a swimmer in a relay race?", "To
swim the fastest", "To hand off the baton", "To dive first", "To flip turn", "To hand off the baton"));
    questions.add(new SportsQuestion("In golf, what is a hole-in-one?", "A score of 1", "A score of 2", "A
score of 3", "A score of 4", "A score of 1"));

    // Additional Questions to reach 50 total
    questions.add(new SportsQuestion("Which sport uses a bat and ball?", "Hockey", "Baseball", "Football",
"Basketball", "Baseball"));
    questions.add(new SportsQuestion("In what sport is the term 'set' commonly used?", "Soccer",
"Tennis", "Basketball", "Baseball", "Tennis"));
    questions.add(new SportsQuestion("Which sport is known as 'the beautiful game'?", "Rugby", "Tennis",
"Soccer", "Basketball", "Soccer"));
    questions.add(new SportsQuestion("What is the main objective in field hockey?", "To score goals", "To
run faster", "To pass the ball", "To defend", "To score goals"));
    questions.add(new SportsQuestion("What is the name of the famous bicycle race held in France?",
"Tour de France", "Paris-Roubaix", "Giro d'Italia", "Vuelta a España", "Tour de France"));
    questions.add(new SportsQuestion("What do we call the area behind home plate in baseball?",
"Infield", "Outfield", "Dugout", "Backstop", "Backstop"));
    questions.add(new SportsQuestion("In basketball, how many fouls does a player get before they are
disqualified?", "4", "5", "6", "3", "5"));
    questions.add(new SportsQuestion("What sport involves a net and is played on a court?", "Football",
"Baseball", "Tennis", "Hockey", "Tennis"));
    questions.add(new SportsQuestion("What do you call a player who catches the ball in American
football?", "Quarterback", "Receiver", "Cornerback", "Linebacker", "Receiver"));
    questions.add(new SportsQuestion("What is the name of the event where runners compete at different
distances?", "Track meet", "Field day", "Olympics", "Marathon", "Track meet"));

    // Shuffle questions for randomness
    Collections.shuffle(questions);
    questions = questions.subList(0, 10); // Get only 10 random questions
  }

  private void displayCurrentQuestion() {
    SportsQuestion currentQuestion = questions.get(currentQuestionIndex);
    questionTextView.setText(currentQuestion.getQuestion());
    questionIndexTextView.setText("Question " + (currentQuestionIndex + 1) + " :");
    optionsGroup.clearCheck();

    // Assuming you have RadioButtons for options
    ((RadioButton) optionsGroup.getChildAt(0)).setText(currentQuestion.getOption1());
    ((RadioButton) optionsGroup.getChildAt(1)).setText(currentQuestion.getOption2());
    ((RadioButton) optionsGroup.getChildAt(2)).setText(currentQuestion.getOption3());
    ((RadioButton) optionsGroup.getChildAt(3)).setText(currentQuestion.getOption4());
  }

  private void showScoreDialog() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Quiz Completed");
    builder.setMessage("Your score is: " + score + " out of " + questions.size());
    builder.setPositiveButton("OK", (dialog, which) -> {
      finish(); // Go back to the previous activity
    });
    builder.setCancelable(false);
    AlertDialog dialog = builder.create();
    dialog.show();
  }
}
```

OUTPUT:



Quiz Application



Maths Quiz

Question 1 :

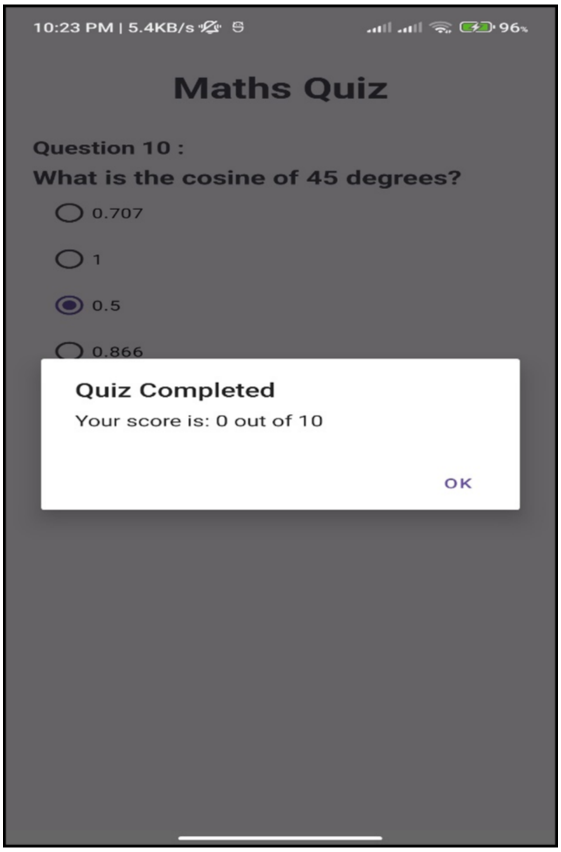
What is 1/2 + 1/4?

○ 3/4

○ 1/2

○ 1/4

○ 1



Maths Quiz

Question 10 :

What is the cosine of 45 degrees?

○ 0.707

○ 1

● 0.5

○ 0.866

Quiz Completed

Your score is: 0 out of 10

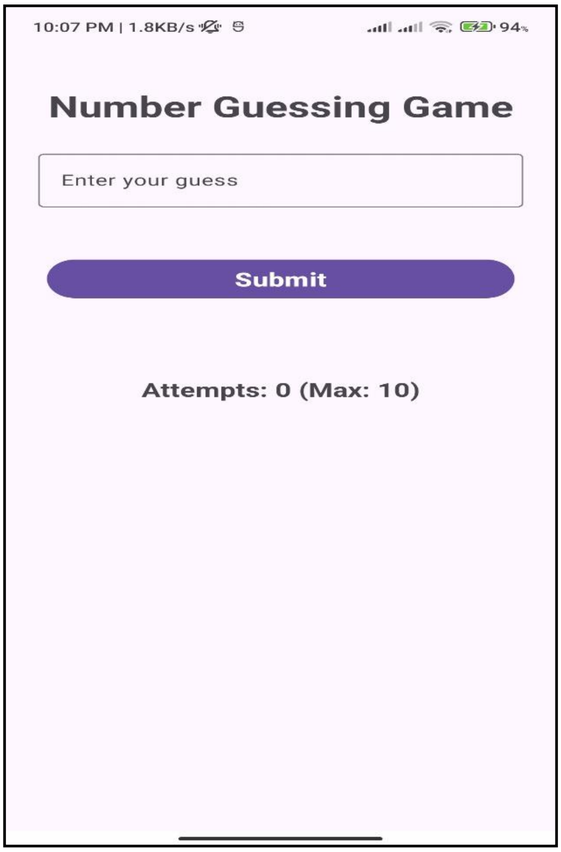
Practical 8

AIM: Number Guessing Game

MainActivity.java

```
package com.example.myapplication;

import android.os.Bundle;
import android.os.Handler; // Import Handler for delays
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import java.util.Random;

public class MainActivity extends AppCompatActivity {

    private int randomNumber;
    private int attempts;
    private static final int MAX_ATTEMPTS = 10; // Maximum number of attempts
    private EditText guessEditText;
    private TextView feedbackTextView, attemptsTextView;

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

        guessEditText = findViewById(R.id.guessEditText);
        feedbackTextView = findViewById(R.id.feedbackTextView);
        attemptsTextView = findViewById(R.id.attemptsTextView);
        Button submitButton = findViewById(R.id.submitButton);

        resetGame(); // Start a new game

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        submitButton.setOnClickListener(v -> handleGuess());
    }
    private void resetGame() {
        randomNumber = new Random().nextInt(100) + 1; // Random number between 1 and 100
        attempts = 0;
        attemptsTextView.setText("Attempts: 0 (Max: 10)");
        feedbackTextView.setText("");
        guessEditText.setText("");
    }

    private void handleGuess() {
        String guessInput = guessEditText.getText().toString();

        if (!guessInput.isEmpty()) {
            int guess = Integer.parseInt(guessInput);
            attempts++;
            attemptsTextView.setText("Attempts: " + attempts);

            if (guess < randomNumber) {
                feedbackTextView.setText("Too low! Try again.");
            } else if (guess > randomNumber) {
                feedbackTextView.setText("Too high! Try again.");
            } else {
                feedbackTextView.setText("Congratulations! You guessed it!");
                // Delay reset game for 2 seconds
                delayResetGame(2000);
                return; // Prevent further logic execution on win
            }

            if (attempts >= MAX_ATTEMPTS) {
                feedbackTextView.setText("Game Over! The number was " + randomNumber + ". Try again!");
                // Delay reset game for 2 seconds
                delayResetGame(2000);
                return; // Prevent further logic execution on game over
            }

            // Clear the EditText after 1 second
            clearEditTextAfterDelay(500);
        }
    }

    private void delayResetGame(int delayMillis) {
        new Handler().postDelayed(this::resetGame, delayMillis);
    }

    private void clearEditTextAfterDelay(int delayMillis) {
        new Handler().postDelayed(() -> guessEditText.setText(""), delayMillis);
    }
}
```

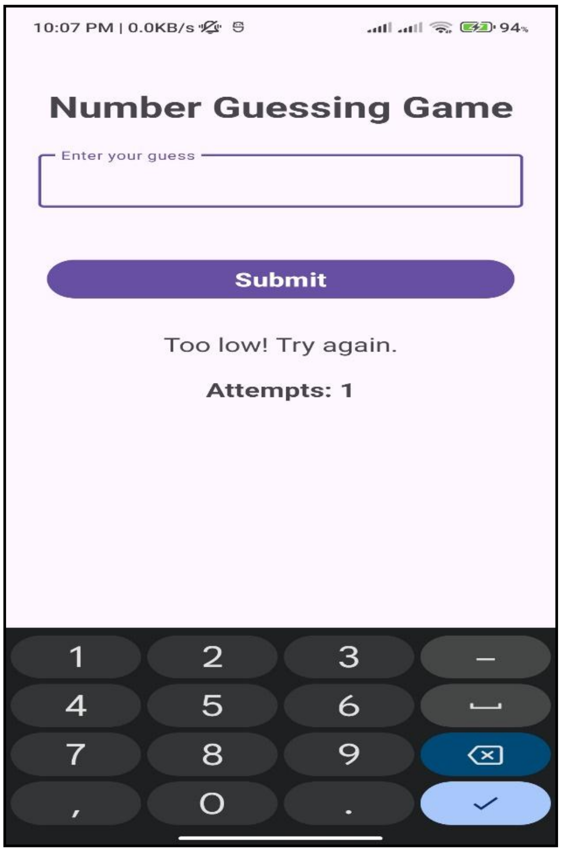
OUTPUT:



Number Guessing Game

Enter your guess

Attempts: 0 (Max: 10)

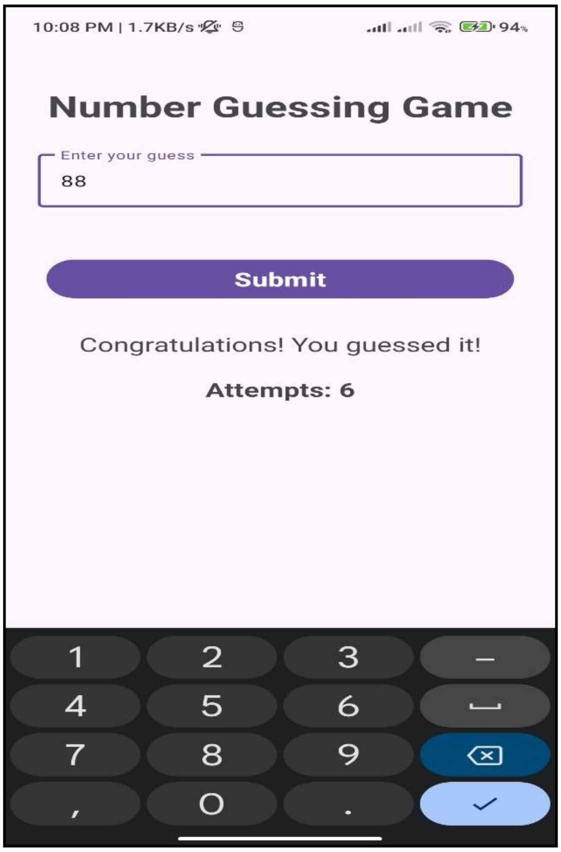


Number Guessing Game

Enter your guess

Too low! Try again.

Attempts: 1

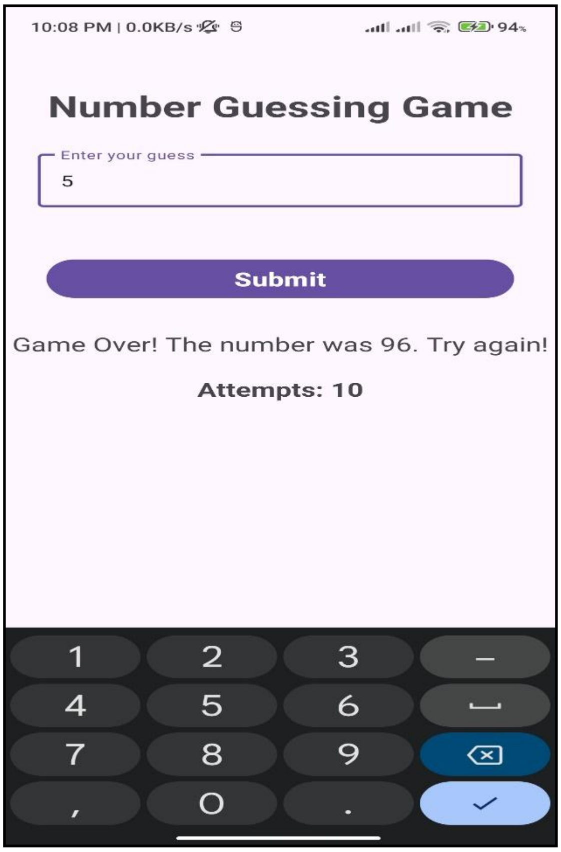


Number Guessing Game

Enter your guess
88

Congratulations! You guessed it!

Attempts: 6



Number Guessing Game

Enter your guess
5

Game Over! The number was 96. Try again!

Attempts: 10

Practical 9

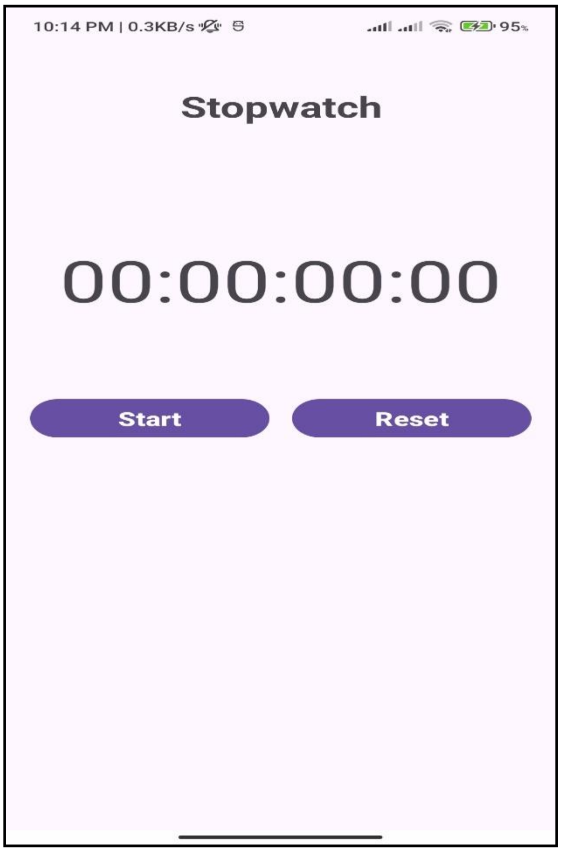
AIM: Stopwatch

MainActivity.java

```
package com.example.myapplication;

import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    private TextView tvTime;
    private Button buttonStartStop;
    private Button buttonReset;

    private Handler handler = new Handler();
    private long startTime = 0;
    private long elapsedTime = 0; // Variable to hold elapsed time
    private boolean isRunning = false;

    private Runnable updateTimeRunnable = new Runnable() {
        @Override
        public void run() {
            if (isRunning) {
                long currentTime = System.currentTimeMillis();
                elapsedTime = currentTime - startTime; // Calculate elapsed time

                // Calculate hours, minutes, seconds, and milliseconds
                int milliseconds = (int) (elapsedTime % 1000) / 10; // Milliseconds to two decimal places
                int seconds = (int) (elapsedTime / 1000) % 60;
                int minutes = (int) ((elapsedTime / (1000 * 60)) % 60);
                int hours = (int) ((elapsedTime / (1000 * 60 * 60)) % 24);

                // Format the time to include milliseconds
                tvTime.setText(String.format("%02d:%02d:%02d.%02d", hours, minutes, seconds, milliseconds));

                handler.postDelayed(this, 10); // Update every 10 milliseconds
            }
        }
    };
```

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

    tvTime = findViewById(R.id.tv_time);
    buttonStartStop = findViewById(R.id.button_start_stop);
    buttonReset = findViewById(R.id.button_reset);

    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.stopwatch_layout), (v, insets) -> {
      Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
      v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
      return insets;
    });

    buttonStartStop.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        if (isRunning) {
          // Stop the stopwatch and save the elapsed time
          isRunning = false;
          buttonStartStop.setText("Start");
          // Calculate the elapsed time when stopped
          elapsedTime = System.currentTimeMillis() - startTime;
        } else {
          // Start from the previously stored elapsed time
          startTime = System.currentTimeMillis() - elapsedTime; // Adjust start time to resume
          isRunning = true;
          buttonStartStop.setText("Stop");
          handler.post(updateTimeRunnable);
        }
      }
    });

    buttonReset.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        // Reset the stopwatch
        isRunning = false;
        startTime = 0;
        elapsedTime = 0; // Reset elapsed time
        tvTime.setText("00:00:00.00"); // Update format to include milliseconds
        buttonStartStop.setText("Start");
        handler.removeCallbacks(updateTimeRunnable);
      }
    });
  }
}
```

OUTPUT:



10:14 PM | 0.3KB/s

95%

Stopwatch

00:00:00:00

Practical 10

AIM: Theme Application for Wallpaper and Background Color Change.

ColorActivity.java

```
package com.example.myapplication;

import android.graphics.Color;
import android.os.Bundle;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class ColorActivity extends AppCompatActivity {

   private ImageView previewView;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      EdgeToEdge.enable(this);
      setContentView(R.layout.activity_color);
      ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
         Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
         v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
         return insets;
      });

      previewView = findViewById(R.id.previewView);
      Button changeBackgroundColorButton = findViewById(R.id.changeBackgroundColorButton);

      changeBackgroundColorButton.setOnClickListener(v -> {
         int color = getRandomColor();
         previewView.setBackgroundColor(color);
         Toast.makeText(ColorActivity.this, "Background color changed", Toast.LENGTH_SHORT).show();
      });
   }

   private int getRandomColor() {
      return Color.rgb((int) (Math.random() * 256), (int) (Math.random() * 256), (int) (Math.random() * 256));
   }
}
```

MainActivity.java

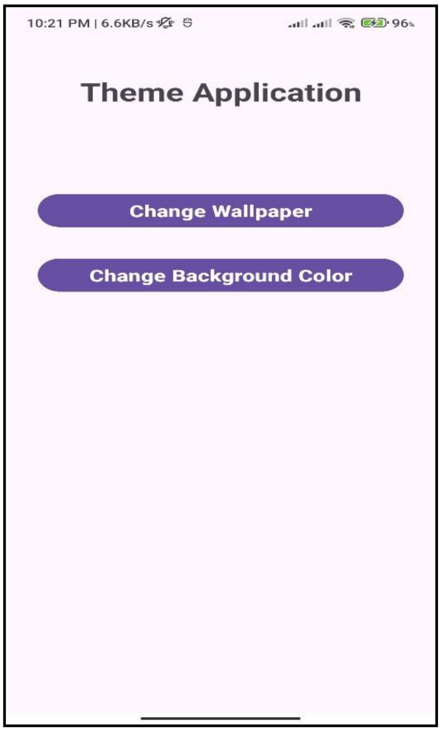
```
package com.example.myapplication;

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

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

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

        // Set up padding for the main view
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        // Find the buttons by their ID
        Button changeWallpaperButton = findViewById(R.id.changeWallpaperButton);
        Button changeBackgroundColorButton = findViewById(R.id.changeBackgroundColorButton);

        // Set onClick listeners for the buttons
        changeWallpaperButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Start the new activity when the button is clicked
                Intent intent = new Intent(MainActivity.this, WallpaperActivity.class);
                startActivity(intent);
            }
        });

        changeBackgroundColorButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Start the new activity when the button is clicked
                Intent intent = new Intent(MainActivity.this, ColorActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

WallerpaperActivity.java

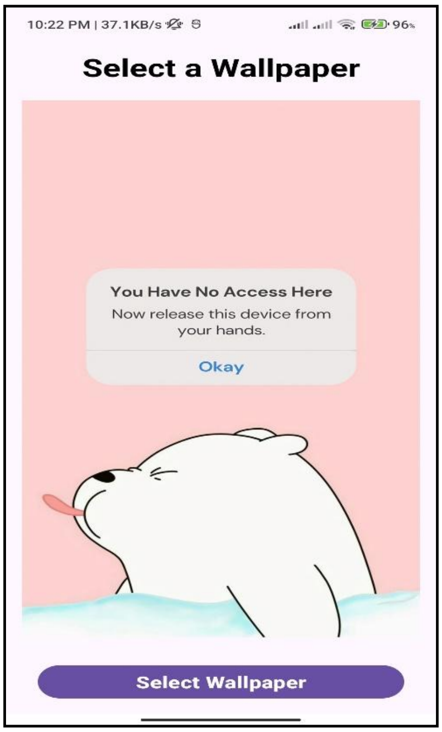
```
package com.example.myapplication;

import android.Manifest; // Import missing
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import com.bumptech.glide.Glide;

public class WallpaperActivity extends AppCompatActivity {

    private static final int PICK_IMAGE = 1;
    private static final int REQUEST_CODE_READ_EXTERNAL_STORAGE = 100;
    private ImageView backgroundImageView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_wallpaper); // Update to your correct layout name

        backgroundImageView = findViewById(R.id.backgroundImageView);
        Button changeWallpaperButton = findViewById(R.id.changeWallpaperButton);

        // Setting up the window insets
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        // Change wallpaper button functionality
        changeWallpaperButton.setOnClickListener(v -> {
```

```
      if (ContextCompat.checkSelfPermission(WallpaperActivity.this,
Manifest.permission.READ_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(WallpaperActivity.this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE}, REQUEST_CODE_READ_EXTERNAL_STORAGE);
      } else {
        openImagePicker(); // Open the image picker if permission is granted
      }
    });
  }

  // Method to open image picker
  private void openImagePicker() {
    Intent intent = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, PICK_IMAGE);
  }

  // Handle the result of the image picker
  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == PICK_IMAGE && resultCode == RESULT_OK && data != null) {
      Uri selectedImageUri = data.getData();
      if (selectedImageUri != null) {
        // Use Glide to load the selected image into the backgroundImageView
        Glide.with(this)
            .load(selectedImageUri)
            .into(backgroundImageView);
        Toast.makeText(this, "Wallpaper changed", Toast.LENGTH_SHORT).show();
      }
    }
  }

  // Handle the permission request result
  @Override
  public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == REQUEST_CODE_READ_EXTERNAL_STORAGE) {
      if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
        openImagePicker(); // Open the image picker if permission is granted
      } else {
        Toast.makeText(this, "Permission denied to read external storage", Toast.LENGTH_SHORT).show();
      }
    }
  }
}
```

OUTPUT:


10:21 PM | 6.6KB/s 96%

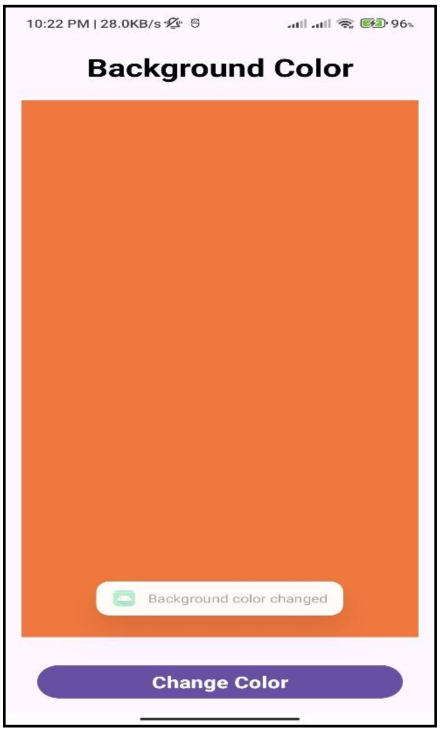
Theme Application


10:22 PM | 37.1KB/s 96%

Select a Wallpaper

You Have No Access Here

Now release this device from your hands.


10:22 PM | 28.0KB/s 96%

Background Color

Background color changed