**13. File I/O Develop a user registration application which data from user and insert received data in a text file (generated/existing) on the same device.**

Main.xml

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

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

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

android:orientation="vertical"

android:padding="20dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<EditText

android:id="@+id/editTextName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Enter Name"

android:minHeight="48dp"

tools:ignore="HardcodedText,TextFields,VisualLintTextFieldSize"

android:autofillHints="" />

<EditText

android:id="@+id/editTextEmail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Enter Email"

android:inputType="textEmailAddress"

android:minHeight="48dp"

tools:ignore="HardcodedText,VisualLintTextFieldSize"

android:autofillHints="" />

<EditText

android:id="@+id/editTextPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:autofillHints=""

android:hint="Enter Password"

android:inputType="textPassword"

android:minHeight="48dp"

tools:ignore="HardcodedText,VisualLintTextFieldSize,TextContrastCheck" />

<Button

android:id="@+id/btnRegister"

android:text="Register"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

tools:ignore="HardcodedText" />

<TextView

android:text="Registered Users:"

android:textSize="18sp"

android:layout\_marginTop="20dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

tools:ignore="HardcodedText" />

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recyclerViewUsers"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"/>

</LinearLayout>

user\_item.xml

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

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

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

android:orientation="vertical"

android:padding="10dp"

android:background="#EEEEEE"

android:layout\_margin="5dp"

android:layout\_width="match\_parent"

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

<TextView

android:id="@+id/txtName"

android:text="Name"

android:textSize="16sp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

tools:ignore="HardcodedText" />

<TextView

android:id="@+id/txtEmail"

android:text="Email"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

tools:ignore="HardcodedText" />

<TextView

android:id="@+id/txtPassword"

android:text="Password"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

tools:ignore="HardcodedText" />

<LinearLayout

android:orientation="horizontal"

android:layout\_width="wrap\_content"

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

<Button

android:id="@+id/btnEdit"

android:text="Edit"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

tools:ignore="ButtonStyle,HardcodedText" />

<Button

android:id="@+id/btnDelete"

android:text="Delete"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

tools:ignore="ButtonStyle,HardcodedText" />

</LinearLayout>

</LinearLayout>

ActivityMain.java

package com.example.mca\_p13;

import android.annotation.SuppressLint;

import android.os.Bundle;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import java.io.BufferedReader;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity implements UserAdapter.UserActionListener {

EditText editTextName, editTextEmail, editTextPassword;

Button btnRegister;

RecyclerView recyclerViewUsers;

String fileName = "user\_data.txt";

ArrayList<User> usersList;

UserAdapter adapter;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

editTextName = findViewById(R.id.editTextName);

editTextEmail = findViewById(R.id.editTextEmail);

editTextPassword = findViewById(R.id.editTextPassword);

btnRegister = findViewById(R.id.btnRegister);

recyclerViewUsers = findViewById(R.id.recyclerViewUsers);

usersList = new ArrayList<>();

adapter = new UserAdapter(this, usersList, this);

recyclerViewUsers.setLayoutManager(new LinearLayoutManager(this));

recyclerViewUsers.setAdapter(adapter);

loadUsersFromFile();

btnRegister.setOnClickListener(v -> saveUserData());

}

@SuppressLint("NotifyDataSetChanged")

private void saveUserData() {

String name = editTextName.getText().toString().trim();

String email = editTextEmail.getText().toString().trim();

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

if (name.isEmpty() || email.isEmpty() || password.isEmpty()) {

Toast.makeText(this, "Please fill all fields", Toast.LENGTH\_SHORT).show();

return;

}

User newUser = new User(name, email, password);

usersList.add(newUser);

adapter.notifyDataSetChanged();

saveAllUsersToFile();

editTextName.setText("");

editTextEmail.setText("");

editTextPassword.setText("");

Toast.makeText(this, "User registered successfully!", Toast.LENGTH\_SHORT).show();

}

private void saveAllUsersToFile() {

try {

FileOutputStream fos = openFileOutput(fileName, MODE\_PRIVATE);

OutputStreamWriter osw = new OutputStreamWriter(fos);

for (User u : usersList) {

osw.write(u.toString() + "\n");

}

osw.close();

} catch (Exception e) {

Toast.makeText(this, "Error saving file", Toast.LENGTH\_SHORT).show();

}

}

@SuppressLint("NotifyDataSetChanged")

private void loadUsersFromFile() {

try {

FileInputStream fis = openFileInput(fileName);

BufferedReader br = new BufferedReader(new InputStreamReader(fis));

String line;

while ((line = br.readLine()) != null) {

User user = User.fromString(line);

if (user != null) usersList.add(user);

}

br.close();

adapter.notifyDataSetChanged();

} catch (FileNotFoundException e) {

// First time, ignore

} catch (IOException e) {

Toast.makeText(this, "Error reading file", Toast.LENGTH\_SHORT).show();

}

}

@SuppressLint("NotifyDataSetChanged")

@Override

public void onDelete(User user) {

usersList.remove(user);

adapter.notifyDataSetChanged();

saveAllUsersToFile();

Toast.makeText(this, "User deleted", Toast.LENGTH\_SHORT).show();

}

@SuppressLint("NotifyDataSetChanged")

@Override

public void onEdit(User user, int position) {

editTextName.setText(user.name);

editTextEmail.setText(user.email);

editTextPassword.setText(user.password);

// Remove old entry (it will be added again after update)

usersList.remove(position);

adapter.notifyDataSetChanged();

saveAllUsersToFile();

}

}

User.java

package com.example.mca\_p13;

import androidx.annotation.NonNull;

public class User {

String name, email, password;

public User(String name, String email, String password) {

this.name = name;

this.email = email;

this.password = password;

}

@NonNull

@Override

public String toString() {

return name + "," + email + "," + password;

}

public static User fromString(String line) {

String[] parts = line.split(",");

if (parts.length == 3) {

return new User(parts[0], parts[1], parts[2]);

}

return null;

}

}

UserAdapter.java

package com.example.mca\_p13;

import android.app.AlertDialog;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

public class UserAdapter extends RecyclerView.Adapter<UserAdapter.UserViewHolder> {

Context context;

ArrayList<User> userList;

UserActionListener listener;

public interface UserActionListener {

void onDelete(User user);

void onEdit(User user, int position);

}

public UserAdapter(Context context, ArrayList<User> userList, UserActionListener listener) {

this.context = context;

this.userList = userList;

this.listener = listener;

}

@NonNull

@Override

public UserViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View v = LayoutInflater.from(context).inflate(R.layout.user\_item, parent, false);

return new UserViewHolder(v);

}

@Override

public void onBindViewHolder(@NonNull UserViewHolder holder, int position) {

User user = userList.get(position);

holder.txtName.setText(user.name);

holder.txtEmail.setText(user.email);

holder.txtPassword.setText(user.password);

holder.btnDelete.setOnClickListener(v -> new AlertDialog.Builder(context)

.setTitle("Delete User")

.setMessage("Are you sure you want to delete this user?")

.setPositiveButton("Yes", (dialog, which) -> listener.onDelete(user))

.setNegativeButton("No", null)

.show());

holder.btnEdit.setOnClickListener(v -> listener.onEdit(user, position));

}

@Override

public int getItemCount() {

return userList.size();

}

public static class UserViewHolder extends RecyclerView.ViewHolder {

TextView txtName, txtEmail, txtPassword;

Button btnDelete, btnEdit;

public UserViewHolder(@NonNull View itemView) {

super(itemView);

txtName = itemView.findViewById(R.id.txtName);

txtEmail = itemView.findViewById(R.id.txtEmail);

txtPassword = itemView.findViewById(R.id.txtPassword);

btnDelete = itemView.findViewById(R.id.btnDelete);

btnEdit = itemView.findViewById(R.id.btnEdit);

}

}

}

--------------------------------------------------------------------------------------------------------------------

**14. Shared Preferences Develop an application that asks the user to select background color of application. The application must remember and apply selected background color every time when user launches the application then after.**

Main.xml

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

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

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

android:id="@+id/mainLayout"

android:orientation="vertical"

android:gravity="center"

android:padding="20dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:text="Select Background Color"

android:textSize="20sp"

android:layout\_marginBottom="20dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

tools:ignore="HardcodedText" />

<Button

android:id="@+id/btnRed"

android:text="Red"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

tools:ignore="HardcodedText" />

<Button

android:id="@+id/btnGreen"

android:text="Green"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

tools:ignore="HardcodedText" />

<Button

android:id="@+id/btnBlue"

android:text="Blue"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

tools:ignore="HardcodedText" />

</LinearLayout>

ActivityMain.java

package com.example.mca\_p14;

import android.content.SharedPreferences;

import android.graphics.Color;

import android.os.Bundle;

import android.widget.Button;

import android.widget.LinearLayout;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

LinearLayout mainLayout;

Button btnRed, btnGreen, btnBlue;

SharedPreferences sharedPreferences;

String PREF\_NAME = "ColorPref";

String KEY\_COLOR = "backgroundColor";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mainLayout = findViewById(R.id.mainLayout);

btnRed = findViewById(R.id.btnRed);

btnGreen = findViewById(R.id.btnGreen);

btnBlue = findViewById(R.id.btnBlue);

// Load SharedPreferences

sharedPreferences = getSharedPreferences(PREF\_NAME, MODE\_PRIVATE);

String savedColor = sharedPreferences.getString(KEY\_COLOR, "WHITE"); // default White

// Apply saved color

applyColor(savedColor);

// Button Listeners

btnRed.setOnClickListener(v -> saveAndApplyColor("RED"));

btnGreen.setOnClickListener(v -> saveAndApplyColor("GREEN"));

btnBlue.setOnClickListener(v -> saveAndApplyColor("BLUE"));

}

private void saveAndApplyColor(String color) {

// Save in SharedPreferences

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putString(KEY\_COLOR, color);

editor.apply();

// Apply color immediately

applyColor(color);

}

private void applyColor(String color) {

switch (color) {

case "RED":

mainLayout.setBackgroundColor(Color.RED);

break;

case "GREEN":

mainLayout.setBackgroundColor(Color.GREEN);

break;

case "BLUE":

mainLayout.setBackgroundColor(Color.BLUE);

break;

default:

mainLayout.setBackgroundColor(Color.WHITE);

}

}

}

--------------------------------------------------------------------------------------------------------------------

15. Database CRUD Operations Develop a data oriented application for user data (user id, name, address, contact number)management. The application should provide interface to add, update, delete and list data of user (s). (Hint: use SQLite

database).

Main.xml

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

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

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

android:orientation="vertical"

android:padding="16dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<EditText

android:id="@+id/etId"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:enabled="false"

android:hint="User ID (AutoFill)"

android:inputType="number"

android:minHeight="48dp"

tools:ignore="HardcodedText,VisualLintTextFieldSize"

android:autofillHints="" />

<EditText

android:id="@+id/etName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Name"

android:minHeight="48dp"

android:autofillHints=""

tools:ignore="HardcodedText,TextFields,VisualLintTextFieldSize" />

<EditText

android:id="@+id/etAddress"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Address"

android:minHeight="48dp"

tools:ignore="HardcodedText,TextFields,VisualLintTextFieldSize"

android:autofillHints="" />

<EditText

android:id="@+id/etContact"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Contact"

android:inputType="phone"

android:minHeight="48dp"

android:autofillHints=""

tools:ignore="HardcodedText,VisualLintTextFieldSize" />

<LinearLayout

android:orientation="horizontal"

android:gravity="center"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp">

<Button

android:id="@+id/btnAdd"

android:text="Add"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

tools:ignore="ButtonStyle,HardcodedText" />

<Button

android:id="@+id/btnUpdate"

android:text="Update"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

tools:ignore="ButtonStyle,HardcodedText" />

<Button

android:id="@+id/btnDelete"

android:text="Delete"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

tools:ignore="ButtonStyle,HardcodedText" />

</LinearLayout>

<Button

android:id="@+id/btnList"

android:text="List Users"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

tools:ignore="HardcodedText,VisualLintButtonSize" />

<ListView

android:id="@+id/listViewUsers"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp"/>

</LinearLayout>

ActivityMain.java

package com.example.mca\_p15;

import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor;

import android.os.Bundle;

import android.widget.\*;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

EditText etId, etName, etAddress, etContact;

Button btnAdd, btnUpdate, btnDelete, btnList;

ListView listViewUsers;

DBHelper dbHelper;

ArrayAdapter<String> adapter;

ArrayList<String> userList;

ArrayList<Integer> userIdList;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

etId = findViewById(R.id.etId);

etName = findViewById(R.id.etName);

etAddress = findViewById(R.id.etAddress);

etContact = findViewById(R.id.etContact);

btnAdd = findViewById(R.id.btnAdd);

btnUpdate = findViewById(R.id.btnUpdate);

btnDelete = findViewById(R.id.btnDelete);

btnList = findViewById(R.id.btnList);

listViewUsers = findViewById(R.id.listViewUsers);

dbHelper = new DBHelper(this);

userList = new ArrayList<>();

userIdList = new ArrayList<>();

// ADD

btnAdd.setOnClickListener(v -> {

boolean inserted = dbHelper.insertUser(

etName.getText().toString(),

etAddress.getText().toString(),

etContact.getText().toString()

);

if (inserted) {

Toast.makeText(this, "User Added", Toast.LENGTH\_SHORT).show();

clearFields();

loadUsers();

} else {

Toast.makeText(this, "Error Adding User", Toast.LENGTH\_SHORT).show();

}

});

// UPDATE

btnUpdate.setOnClickListener(v -> {

boolean updated = dbHelper.updateUser(

etId.getText().toString(),

etName.getText().toString(),

etAddress.getText().toString(),

etContact.getText().toString()

);

if (updated) {

Toast.makeText(this, "User Updated", Toast.LENGTH\_SHORT).show();

clearFields();

loadUsers();

} else {

Toast.makeText(this, "Error Updating User", Toast.LENGTH\_SHORT).show();

}

});

// DELETE

btnDelete.setOnClickListener(v -> {

boolean deleted = dbHelper.deleteUser(etId.getText().toString());

if (deleted) {

Toast.makeText(this, "User Deleted", Toast.LENGTH\_SHORT).show();

clearFields();

loadUsers();

} else {

Toast.makeText(this, "Error Deleting User", Toast.LENGTH\_SHORT).show();

}

});

// LIST USERS

btnList.setOnClickListener(v -> loadUsers());

// CLICK ITEM → SHOW IN EDITTEXT

listViewUsers.setOnItemClickListener((parent, view, position, id) -> {

int userId = userIdList.get(position);

Cursor cursor = dbHelper.getAllUsers();

if (cursor != null) {

while (cursor.moveToNext()) {

if (cursor.getInt(0) == userId) {

etId.setText(String.valueOf(cursor.getInt(0)));

etName.setText(cursor.getString(1));

etAddress.setText(cursor.getString(2));

etContact.setText(cursor.getString(3));

break;

}

}

cursor.close();

}

});

}

private void loadUsers() {

userList.clear();

userIdList.clear();

Cursor cursor = dbHelper.getAllUsers();

if (cursor != null) {

while (cursor.moveToNext()) {

int id = cursor.getInt(0);

String name = cursor.getString(1);

String address = cursor.getString(2);

String contact = cursor.getString(3);

userList.add(id + " - " + name + " - "+ address + " - " + contact);

userIdList.add(id);

}

cursor.close();

}

adapter = new ArrayAdapter<>(this, android.R.layout.simple\_list\_item\_1, userList);

listViewUsers.setAdapter(adapter);

}

private void clearFields() {

etId.setText("");

etName.setText("");

etAddress.setText("");

etContact.setText("");

}

}

DBHelper.java

package com.example.mca\_p15;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

public class DBHelper extends SQLiteOpenHelper {

public static final String DATABASE\_NAME = "UserDB.db";

public static final String TABLE\_NAME = "users";

public static final String COL\_ID = "id";

public static final String COL\_NAME = "name";

public static final String COL\_ADDRESS = "address";

public static final String COL\_CONTACT = "contact";

public DBHelper(Context context) {

super(context, DATABASE\_NAME, null, 1);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL("CREATE TABLE " + TABLE\_NAME + " (" +

COL\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +

COL\_NAME + " TEXT, " +

COL\_ADDRESS + " TEXT, " +

COL\_CONTACT + " TEXT)");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NAME);

onCreate(db);

}

public boolean insertUser(String name, String address, String contact) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues cv = new ContentValues();

cv.put(COL\_NAME, name);

cv.put(COL\_ADDRESS, address);

cv.put(COL\_CONTACT, contact);

long result = db.insert(TABLE\_NAME, null, cv);

return result != -1;

}

public boolean updateUser(String id, String name, String address, String contact) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues cv = new ContentValues();

cv.put(COL\_NAME, name);

cv.put(COL\_ADDRESS, address);

cv.put(COL\_CONTACT, contact);

int result = db.update(TABLE\_NAME, cv, COL\_ID + "=?", new String[]{id});

return result > 0;

}

public boolean deleteUser(String id) {

SQLiteDatabase db = this.getWritableDatabase();

int result = db.delete(TABLE\_NAME, COL\_ID + "=?", new String[]{id});

return result > 0;

}

public Cursor getAllUsers() {

SQLiteDatabase db = this.getReadableDatabase();

return db.rawQuery("SELECT \* FROM " + TABLE\_NAME, null);

}

}

--------------------------------------------------------------------------------------------------------------------

**16.Data Driven Authentication Develop a data driven user authentication application that asks for user id and password to user. After accepting user id and password, application should check whether entered authentication details exists in data table or not. Display appropriate message on screen.**

Main.xml

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

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

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

android:orientation="vertical"

android:padding="20dp"

android:gravity="center"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<EditText

android:id="@+id/etUserId"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="User ID"

android:minHeight="48dp"

tools:ignore="HardcodedText,TextFields,VisualLintTextFieldSize"

android:autofillHints="" />

<EditText

android:id="@+id/etPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="Password"

android:inputType="textPassword"

android:minHeight="48dp"

android:autofillHints=""

tools:ignore="HardcodedText,VisualLintTextFieldSize" />

<Button

android:id="@+id/btnLogin"

android:text="Login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp"

tools:ignore="HardcodedText,VisualLintButtonSize" />

<Button

android:id="@+id/btnRegister"

android:text="Register"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

tools:ignore="HardcodedText,VisualLintButtonSize" />

<TextView

android:id="@+id/tvResult"

android:text=""

android:textSize="18sp"

android:textColor="@android:color/holo\_blue\_dark"

android:layout\_marginTop="20dp"

android:layout\_width="wrap\_content"

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

</LinearLayout>

ActivityMain.java

package com.example.mca\_p16;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.os.Bundle;

import android.widget.\*;

public class MainActivity extends AppCompatActivity {

EditText etUserId, etPassword;

Button btnLogin, btnRegister;

TextView tvResult;

DBHelper dbHelper;

@SuppressLint("SetTextI18n")

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

etUserId = findViewById(R.id.etUserId);

etPassword = findViewById(R.id.etPassword);

btnLogin = findViewById(R.id.btnLogin);

btnRegister = findViewById(R.id.btnRegister);

tvResult = findViewById(R.id.tvResult);

dbHelper = new DBHelper(this);

// LOGIN

btnLogin.setOnClickListener(v -> {

String userid = etUserId.getText().toString();

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

if (dbHelper.checkUser(userid, password)) {

tvResult.setText("✅ Login Successful! Welcome " + userid);

} else {

tvResult.setText("❌ Invalid Credentials");

}

});

// REGISTER

btnRegister.setOnClickListener(v -> {

String userid = etUserId.getText().toString();

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

boolean inserted = dbHelper.insertUser(userid, password);

if (inserted) {

tvResult.setText("✅ User Registered Successfully!");

} else {

tvResult.setText("❌ Registration Failed (UserID already exists)");

}

});

}

}

DBHelper.java

package com.example.mca\_p15;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

public class DBHelper extends SQLiteOpenHelper {

private static final String DATABASE\_NAME = "AuthDB";

private static final int DATABASE\_VERSION = 1;

private static final String TABLE\_USERS = "users";

public DBHelper(Context context) {

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

String createTable = "CREATE TABLE " + TABLE\_USERS + " (" +

"id INTEGER PRIMARY KEY AUTOINCREMENT, " +

"userid TEXT UNIQUE, " +

"password TEXT)";

db.execSQL(createTable);

// Insert default user (for testing)

ContentValues values = new ContentValues();

values.put("userid", "admin");

values.put("password", "1234");

db.insert(TABLE\_USERS, null, values);

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("DROP TABLE IF EXISTS " + TABLE\_USERS);

onCreate(db);

}

// Insert new user

public boolean insertUser(String userid, String password) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues values = new ContentValues();

values.put("userid", userid);

values.put("password", password);

long result = db.insert(TABLE\_USERS, null, values);

return result != -1;

}

// Check login credentials

public boolean checkUser(String userid, String password) {

SQLiteDatabase db = this.getReadableDatabase();

Cursor cursor = db.rawQuery("SELECT \* FROM " + TABLE\_USERS + " WHERE userid=? AND password=?",

new String[]{userid, password});

boolean exists = cursor.getCount() > 0;

cursor.close();

return exists;

}

}

--------------------------------------------------------------------------------------------------------------------

**17. Animation Develop an application that contains a spinner and an image. Spinner should contain names of animations. Upon selecting an animation name, selected animation should be applied on image view.**

Main.xml

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

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center"

android:orientation="vertical"

android:padding="20dp">

<Spinner

android:id="@+id/spinnerAnimations"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:minHeight="48dp" />

<ImageView

android:id="@+id/imageView"

android:layout\_width="209dp"

android:layout\_height="259dp"

android:layout\_marginTop="30dp"

android:src="@drawable/dm"

android:contentDescription="TODO"

tools:ignore="ContentDescription,HardcodedText" />

</LinearLayout>

res/anim/fade.xml

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

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

android:fromAlpha="1.0"

android:toAlpha="0.0"

android:duration="1000"

android:repeatCount="infinite"

android:repeatMode="reverse"/>

res/anim/rotate.xml

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

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

android:fromDegrees="0"

android:toDegrees="360"

android:pivotX="50%"

android:pivotY="50%"

android:duration="1000"

android:repeatCount="infinite"/>

res/anim/Zoom.xml

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

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

android:fromXScale="1.0"

android:toXScale="2.0"

android:fromYScale="1.0"

android:toYScale="2.0"

android:pivotX="50%"

android:pivotY="50%"

android:duration="1000"

android:repeatCount="infinite"

android:repeatMode="reverse"/>

ActivityMain.java

package com.example.mca\_p17;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.\*;

public class MainActivity extends AppCompatActivity {

Spinner spinner;

ImageView imageView;

String[] animations = {"Rotate", "Fade", "Zoom"};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

spinner = findViewById(R.id.spinnerAnimations);

imageView = findViewById(R.id.imageView);

ArrayAdapter<String> adapter = new ArrayAdapter<>(this,

android.R.layout.simple\_spinner\_item, animations);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinner.setAdapter(adapter);

spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, android.view.View view, int position, long id) {

Animation anim = null;

switch (animations[position]) {

case "Rotate":

anim = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.rotate);

break;

case "Fade":

anim = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.fade);

break;

case "Zoom":

anim = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.zoom);

break;

}

if (anim != null) {

imageView.startAnimation(anim);

}

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// Do nothing

}

});

}

}

Steps

1] create main.xml and Activitymain.java and write design and code

2]right click on res folder

3]Go to new and click on Android resource directory

4]Give directory name anim and select resource type anim click on ok

4]Go to drawable

5]Past on image on the folder which you want to display and remain that image file name is same as give in activitymain.java file

--------------------------------------------------------------------------------------------------------------------

**18. Audio Player Develop an audio player application having facilities to start, pause and stop audio playback.**

Main.xml

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

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:gravity="center"

android:padding="20dp">

<TextView

android:id="@+id/tvTitle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Simple Audio Player"

android:textSize="20sp"

android:layout\_marginBottom="30dp"

tools:ignore="HardcodedText" />

<Button

android:id="@+id/btnPlay"

android:layout\_width="200dp"

android:layout\_height="wrap\_content"

android:text="Play"

tools:ignore="HardcodedText" />

<Button

android:id="@+id/btnPause"

android:layout\_width="200dp"

android:layout\_height="wrap\_content"

android:text="Pause"

android:layout\_marginTop="20dp"

tools:ignore="HardcodedText" />

<Button

android:id="@+id/btnStop"

android:layout\_width="200dp"

android:layout\_height="wrap\_content"

android:text="Stop"

android:layout\_marginTop="20dp"

tools:ignore="HardcodedText" />

</LinearLayout>

ActivityMain.java

package com.example.mca\_p18;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.media.MediaPlayer;

import android.os.Bundle;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

Button btnPlay, btnPause, btnStop;

MediaPlayer mediaPlayer;

boolean isPaused = false; // to track pause state

@SuppressLint("SetTextI18n")

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

btnPlay = findViewById(R.id.btnPlay);

btnPause = findViewById(R.id.btnPause);

btnStop = findViewById(R.id.btnStop);

// Load audio file from raw folder

mediaPlayer = MediaPlayer.create(this, R.raw.doremon);

// Play Button

btnPlay.setOnClickListener(v -> {

if (!mediaPlayer.isPlaying()) {

mediaPlayer.start();

isPaused = false;

btnPause.setText("Pause"); // reset button text

Toast.makeText(this, "Playing Audio", Toast.LENGTH\_SHORT).show();

}

});

// Pause/Resume Button

btnPause.setOnClickListener(v -> {

if (mediaPlayer.isPlaying()) {

mediaPlayer.pause();

isPaused = true;

btnPause.setText("Resume"); // change text

Toast.makeText(this, "Audio Paused", Toast.LENGTH\_SHORT).show();

} else if (isPaused) {

mediaPlayer.start();

isPaused = false;

btnPause.setText("Pause"); // change back

Toast.makeText(this, "Audio Resumed", Toast.LENGTH\_SHORT).show();

}

});

// Stop Button

btnStop.setOnClickListener(v -> {

if (mediaPlayer.isPlaying() || isPaused) {

mediaPlayer.stop();

mediaPlayer = MediaPlayer.create(this, R.raw.doremon); // Reset

isPaused = false;

btnPause.setText("Pause"); // reset text

Toast.makeText(this, "Audio Stopped", Toast.LENGTH\_SHORT).show();

}

});

}

@Override

protected void onDestroy() {

if (mediaPlayer != null) {

mediaPlayer.release();

mediaPlayer = null;

}

super.onDestroy();

}

}

steps to create raw directory

1]right click on res folder

2]Go to new and click on Android resource directory

3]Give directory name raw and select resource type raw click on ok

4]Download a song or audio in mp4 extension

5]Past on raw folder

**19. Web Browser Develop an application to show contents of specified URL without using native browser. Also provide facility to navigate to previous and next page as well as clear browsing history.**

Main.xml

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

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

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

android:id="@+id/root"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<!-- URL bar + Go -->

<LinearLayout

android:orientation="horizontal"

android:padding="8dp"

android:layout\_width="match\_parent"

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

<EditText

android:id="@+id/etUrl"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:autofillHints=""

android:hint="Enter URL e.g. https://example.com"

android:inputType="textUri"

android:minHeight="48dp"

android:singleLine="true"

tools:ignore="HardcodedText,VisualLintTextFieldSize,TextContrastCheck" />

<Button

android:id="@+id/btnGo"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Go"

tools:ignore="HardcodedText,TouchTargetSizeCheck" />

</LinearLayout>

<!-- Controls -->

<LinearLayout

android:orientation="horizontal"

android:paddingStart="8dp"

android:paddingEnd="8dp"

android:paddingBottom="8dp"

android:layout\_width="match\_parent"

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

<Button

android:id="@+id/btnBack"

android:text="◀ Back"

android:layout\_width="0dp"

android:layout\_weight="1"

android:layout\_height="wrap\_content"

tools:ignore="ButtonStyle,HardcodedText,VisualLintButtonSize" />

<Button

android:id="@+id/btnForward"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Next ▶"

tools:ignore="ButtonStyle,HardcodedText,VisualLintButtonSize" />

<Button

android:id="@+id/btnReload"

android:text="Reload"

android:layout\_width="0dp"

android:layout\_weight="1"

android:layout\_height="wrap\_content"

tools:ignore="ButtonStyle,HardcodedText,VisualLintButtonSize" />

<Button

android:id="@+id/btnClearHistory"

android:text="Clear "

android:layout\_width="0dp"

android:layout\_weight="1"

android:layout\_height="wrap\_content"

tools:ignore="ButtonStyle,HardcodedText,VisualLintButtonSize" />

</LinearLayout>

<!-- Tiny progress bar -->

<ProgressBar

android:id="@+id/progress"

style="?android:attr/progressBarStyleHorizontal"

android:max="100"

android:visibility="gone"

android:layout\_width="match\_parent"

android:layout\_height="3dp" />

<!-- Web content -->

<WebView

android:id="@+id/webView"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1"/>

</LinearLayout>

ActivityMain.java

package com.example.mca\_p19;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.graphics.Bitmap;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.view.inputmethod.EditorInfo;

import android.webkit.CookieManager;

import android.webkit.WebChromeClient;

import android.webkit.WebSettings;

import android.webkit.WebView;

import android.webkit.WebViewClient;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ProgressBar;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

private WebView webView;

private EditText etUrl;

private Button btnBack;

private Button btnForward;

private ProgressBar progress;

@SuppressLint("SetJavaScriptEnabled")

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

webView = findViewById(R.id.webView);

etUrl = findViewById(R.id.etUrl);

Button btnGo = findViewById(R.id.btnGo);

btnBack = findViewById(R.id.btnBack);

btnForward = findViewById(R.id.btnForward);

Button btnReload = findViewById(R.id.btnReload);

Button btnClearHistory = findViewById(R.id.btnClearHistory);

progress = findViewById(R.id.progress);

// WebView settings

WebSettings ws = webView.getSettings();

ws.setJavaScriptEnabled(true);

ws.setDomStorageEnabled(true);

ws.setLoadWithOverviewMode(true);

ws.setUseWideViewPort(true);

ws.setBuiltInZoomControls(true);

ws.setDisplayZoomControls(false);

webView.setWebViewClient(new WebViewClient() {

@Override public void onPageStarted(WebView view, String url, Bitmap favicon) {

progress.setVisibility(View.VISIBLE);

etUrl.setText(url);

updateNavButtons();

}

@Override public void onPageFinished(WebView view, String url) {

progress.setVisibility(View.GONE);

updateNavButtons();

}

});

webView.setWebChromeClient(new WebChromeClient() {

@Override public void onProgressChanged(WebView view, int newProgress) {

progress.setProgress(newProgress);

if (newProgress >= 100) progress.setVisibility(View.GONE);

}

});

// Actions

btnGo.setOnClickListener(v -> loadFromBar());

etUrl.setOnEditorActionListener((v, actionId, event) -> {

if (actionId == EditorInfo.IME\_ACTION\_GO || actionId == EditorInfo.IME\_ACTION\_DONE) {

loadFromBar();

return true;

}

return false;

});

btnBack.setOnClickListener(v -> {

if (webView.canGoBack()) webView.goBack();

updateNavButtons();

});

btnForward.setOnClickListener(v -> {

if (webView.canGoForward()) webView.goForward();

updateNavButtons();

});

btnReload.setOnClickListener(v -> webView.reload());

btnClearHistory.setOnClickListener(v -> {

webView.clearHistory();

webView.clearCache(true);

CookieManager.getInstance().removeAllCookies(null);

CookieManager.getInstance().flush();

updateNavButtons();

Toast.makeText(this, "History & cache cleared", Toast.LENGTH\_SHORT).show();

});

// Load a default page

String home = "https://www.example.com";

etUrl.setText(home);

webView.loadUrl(home);

}

private void loadFromBar() {

String url = etUrl.getText().toString().trim();

if (TextUtils.isEmpty(url)) {

Toast.makeText(this, "Enter a URL", Toast.LENGTH\_SHORT).show();

return;

}

// Normalize: add scheme if missing

if (!url.startsWith("http://") && !url.startsWith("https://")) {

url = "https://" + url;

}

webView.loadUrl(url);

}

private void updateNavButtons() {

btnBack.setEnabled(webView.canGoBack());

btnForward.setEnabled(webView.canGoForward());

}

/\*\*

\*

\*/

// Handle device Back button to navigate WebView history first

@Override

public void onBackPressed() {

if (webView.canGoBack()) {

webView.goBack();

updateNavButtons();

} else {

super.onBackPressed();

}

}

@Override

protected void onDestroy() {

// Clean up WebView to avoid leaks

if (webView != null) {

webView.loadUrl("about:blank");

webView.clearHistory();

((android.view.ViewGroup) webView.getParent()).removeView(webView);

webView.destroy();

}

super.onDestroy();

}

}

Maniestf.xml

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

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

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

<uses-permission android:name="android.permission.INTERNET"/>

<application

android:allowBackup="true"

android:dataExtractionRules="@xml/data\_extraction\_rules"

android:fullBackupContent="@xml/backup\_rules"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/Theme.MCAP19"

tools:targetApi="31">

<activity

android:name=".MainActivity"

android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

**20. Location Based Services Develop an application that keeps track of location (coordinates) of device and display values of longitude and latitude on screen. Printed?**

Main.xml

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

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

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

android:orientation="vertical"

android:padding="16dp"

android:gravity="center"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:id="@+id/locationText"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Fetching location..."

android:textSize="18sp"

android:padding="16dp"

android:textStyle="bold"

tools:ignore="HardcodedText" />

<Button

android:id="@+id/btnGetLocation"

android:text="Get Location"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:padding="12dp"

android:layout\_marginTop="20dp"

tools:ignore="HardcodedText" />

</LinearLayout>

ActivityMain.java

package com.example.mca\_p20;

import android.Manifest;

import android.annotation.SuppressLint;

import android.content.pm.PackageManager;

import android.location.Address;

import android.location.Geocoder;

import android.os.Bundle;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import com.google.android.gms.location.FusedLocationProviderClient;

import com.google.android.gms.location.LocationServices;

import java.io.IOException;

import java.util.List;

import java.util.Locale;

public class MainActivity extends AppCompatActivity {

private static final int REQUEST\_CODE = 101;

private FusedLocationProviderClient fusedLocationClient;

private TextView locationText;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

locationText = findViewById(R.id.locationText);

fusedLocationClient = LocationServices.getFusedLocationProviderClient(this);

getLastLocation();

}

private void getLastLocation() {

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED &&

ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION}, REQUEST\_CODE);

return;

}

fusedLocationClient.getLastLocation().addOnSuccessListener(this, location -> {

if (location != null) {

double lat = location.getLatitude();

double lon = location.getLongitude();

// Default lat/lon display

@SuppressLint({"StringFormatInvalid", "LocalSuppress"}) String locText = getString(R.string.location\_text, lat, lon);

// Now get address

Geocoder geocoder = new Geocoder(MainActivity.this, Locale.getDefault());

try {

List<Address> addresses = geocoder.getFromLocation(lat, lon, 1);

if (addresses != null && !addresses.isEmpty()) {

Address address = addresses.get(0);

String fullAddress = address.getAddressLine(0);

locText += "\nAddress: " + fullAddress;

}

} catch (IOException e) {

e.printStackTrace();

Toast.makeText(MainActivity.this, "Unable to get address", Toast.LENGTH\_SHORT).show();

}

locationText.setText(locText);

} else {

Toast.makeText(MainActivity.this, getString(R.string.location\_not\_found), Toast.LENGTH\_SHORT).show();

}

});

}

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults);

if (requestCode == REQUEST\_CODE && grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

getLastLocation();

} else {

Toast.makeText(this, getString(R.string.permission\_denied), Toast.LENGTH\_SHORT).show();

}

}

}

Manifest.xml

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

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

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

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/>

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"/>

<uses-permission android:name="android.permission.INTERNET"/>

<application

android:allowBackup="true"

android:dataExtractionRules="@xml/data\_extraction\_rules"

android:fullBackupContent="@xml/backup\_rules"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/Theme.MCAP20"

tools:targetApi="31">

<activity

android:name=".MainActivity"

android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>