**MAD – ALL PROGRAMS:**

1. **Basic Android App :**

MAD -> Mobile Application Development

practical - 1

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

1) XML file :-

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

<RelativeLayout

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

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

android:id="@+id/idRLContainer"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".MainActivity">

<!--on below line we are creating

a text view for heading of our app-->

<TextView

android:id="@+id/idTVHeading1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_above="@id/idTVMsg"

android:layout\_centerInParent="true"

android:layout\_margin="20dp"

android:gravity="center"

android:padding="10dp"

android:text="Center Alignment of Text View"

android:textAlignment="center"

android:textColor="@color/black"

android:textSize="20sp"

android:textStyle="bold" />

<!-- on below line we are creating a text view for center horizontally

aligned text view. In the below text view we are specifying

layout\_centerHorizontal to true to align text view center

horizontally to center of screen and we are specifying

layout\_centerVertical to true to align text view center

vertically to center of screen -->

<TextView

android:id="@+id/idTVMsg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_centerVertical="true"

android:padding="4dp"

android:text="Welcome to Geeks for Geeks"

android:textColor="@color/black"

android:textSize="20sp"

android:textStyle="bold" />

</RelativeLayout>

1. **Basic android widgets :**

MAD -> Practical - 2

1) Main Activity.java :-

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

package com.example.sairamkrishna.myapplication;

import android.app.Activity;

import android.graphics.Color;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

public class MainActivity extends Activity {

Button b1,b2;

EditText ed1,ed2;

TextView tx1;

int counter = 3;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

b1 = (Button)findViewById(R.id.button);

ed1 = (EditText)findViewById(R.id.editText);

ed2 = (EditText)findViewById(R.id.editText2);

b2 = (Button)findViewById(R.id.button2);

tx1 = (TextView)findViewById(R.id.textView3);

tx1.setVisibility(View.GONE);

b1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if(ed1.getText().toString().equals("admin") &&

ed2.getText().toString().equals("admin")) {

Toast.makeText(getApplicationContext(),

"Redirecting...",Toast.LENGTH\_SHORT).show();

}else{

Toast.makeText(getApplicationContext(), "Wrong

Credentials",Toast.LENGTH\_SHORT).show();

tx1.setVisibility(View.VISIBLE);

tx1.setBackgroundColor(Color.RED);

counter--;

tx1.setText(Integer.toString(counter));

if (counter == 0) {

b1.setEnabled(false);

}

}

}

});

b2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

finish();

}

});

}

}

2) activity\_main.xml file :-

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

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

<RelativeLayout 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:paddingLeft= "@dimen/activity\_horizontal\_margin"

android:paddingRight = "@dimen/activity\_horizontal\_margin"

android:paddingTop = "@dimen/activity\_vertical\_margin"

android:paddingBottom = "@dimen/activity\_vertical\_margin" tools:context = ".MainActivity">

<TextView android:text = "Login" android:layout\_width="wrap\_content"

android:layout\_height = "wrap\_content"

android:id = "@+id/textview"

android:textSize = "35dp"

android:layout\_alignParentTop = "true"

android:layout\_centerHorizontal = "true" />

<TextView

android:layout\_width = "wrap\_content"

android:layout\_height = "wrap\_content"

android:text = "Tutorials point"

android:id = "@+id/textView"

android:layout\_below = "@+id/textview"

android:layout\_centerHorizontal = "true"

android:textColor = "#ff7aff24"

android:textSize = "35dp" />

<EditText

android:layout\_width = "wrap\_content"

android:layout\_height = "wrap\_content"

android:id = "@+id/editText"

android:hint = "Enter Name"

android:focusable = "true"

android:textColorHighlight = "#ff7eff15"

android:textColorHint = "#ffff25e6"

android:layout\_marginTop = "46dp"

android:layout\_below = "@+id/imageView"

android:layout\_alignParentLeft = "true"

android:layout\_alignParentStart = "true"

android:layout\_alignParentRight = "true"

android:layout\_alignParentEnd = "true" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/imageView"

android:src="@drawable/abc"

android:layout\_below="@+id/textView"

android:layout\_centerHorizontal="true" />

<EditText

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:inputType="textPassword"

android:ems="10"

android:id="@+id/editText2"

android:layout\_below="@+id/editText"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:layout\_alignRight="@+id/editText"

android:layout\_alignEnd="@+id/editText"

android:textColorHint="#ffff299f"

android:hint="Password" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Attempts Left:"

android:id="@+id/textView2"

android:layout\_below="@+id/editText2"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:textSize="25dp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="New Text"

android:id="@+id/textView3"

android:layout\_alignTop="@+id/textView2"

android:layout\_alignParentRight="true"

android:layout\_alignParentEnd="true"

android:layout\_alignBottom="@+id/textView2"

android:layout\_toEndOf="@+id/textview"

android:textSize="25dp"

android:layout\_toRightOf="@+id/textview" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="login"

android:id="@+id/button"

android:layout\_alignParentBottom="true"

android:layout\_toLeftOf="@+id/textview"

android:layout\_toStartOf="@+id/textview" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Cancel"

android:id="@+id/button2"

android:layout\_alignParentBottom="true"

android:layout\_toRightOf="@+id/textview"

android:layout\_toEndOf="@+id/textview" />

</RelativeLayout>

3) androidmanifest.xml file :-

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

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

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

package="com.example.sairamkrishna.myapplication" >

<application

android:allowBackup="true"

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

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

android:theme="@style/AppTheme" >

<activity

android:name=".MainActivity"

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

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

</manifest>

1. **Authentication, Validation and Toast :**

MAD - Practical - 03

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

Here is the full Kotlin code for the authentication application:

1) LoginActivity.kt :

import android.os.Bundle

import android.widget.Button

import android.widget.EditText

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

import android.content.Intent

import android.util.Patterns

class LoginActivity : AppCompatActivity() {

private lateinit var emailEditText: EditText

private lateinit var passwordEditText: EditText

private lateinit var loginButton: Button

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_login)

emailEditText = findViewById(R.id.emailEditText)

passwordEditText = findViewById(R.id.passwordEditText)

loginButton = findViewById(R.id.loginButton)

loginButton.isEnabled = false

emailEditText.setOnKeyListener { \_, \_, \_ ->

loginButton.isEnabled = isValidEmail(emailEditText.text.toString())

false

}

loginButton.setOnClickListener {

val email = emailEditText.text.toString()

val password = passwordEditText.text.toString()

if (authenticateUser(email, password)) {

val welcomeActivityIntent = Intent(this, WelcomeActivity::class.java)

welcomeActivityIntent.putExtra("email", email)

startActivity(welcomeActivityIntent)

} else {

Toast.makeText(this, "Invalid email or password", Toast.LENGTH\_SHORT).show()

}

}

}

private fun isValidEmail(email: String): Boolean {

return Patterns.EMAIL\_ADDRESS.matcher(email).matches()

}

private fun authenticateUser(email: String, password: String): Boolean {

// Replace with your authentication logic

return email == "user@example.com" && password == "password"

}

}

2) WelcomeActivity.kt :

import android.os.Bundle

import android.widget.TextView

import androidx.appcompat.app.AppCompatActivity

class WelcomeActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_welcome)

val email = intent.getStringExtra("email")

val welcomeTextView = findViewById<TextView>(R.id.welcomeTextView)

welcomeTextView.text = "Welcome, $email!"

}

}

1. **Options Menu:**

MAD - Practical - 4

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

Here is the full Kotlin code for the menu-driven application:

MainActivity.kt

import android.graphics.Color

import android.os.Bundle

import android.view.Menu

import android.view.MenuInflater

import android.view.MenuItem

import androidx.appcompat.app.AppCompatActivity

import androidx.core.content.ContextCompat

class MainActivity : AppCompatActivity() {

private lateinit var linearLayout: LinearLayout

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

linearLayout = findViewById(R.id.linearLayout)

}

override fun onCreateOptionsMenu(menu: Menu): Boolean {

val inflater: MenuInflater = menuInflater

inflater.inflate(R.menu.options\_menu, menu)

return true

}

override fun onOptionsItemSelected(item: MenuItem): Boolean {

return when (item.itemId) {

R.id.red -> {

linearLayout.setBackgroundColor(ContextCompat.getColor(this, R.color.red))

true

}

R.id.green -> {

linearLayout.setBackgroundColor(ContextCompat.getColor(this, R.color.green))

true

}

R.id.blue -> {

linearLayout.setBackgroundColor(ContextCompat.getColor(this, R.color.blue))

true

}

else -> super.onOptionsItemSelected(item)

}

}

}

activity\_main.xml

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

<LinearLayout xmlns:android="(link unavailable)"

android:id="@+id/linearLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

</LinearLayout>

options\_menu.xml

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

<menu xmlns:android="(link unavailable)"

xmlns:app="(link unavailable)">

<item

android:id="@+id/red"

android:title="Red"

app:showAsAction="never" />

<item

android:id="@+id/green"

android:title="Green"

app:showAsAction="never" />

<item

android:id="@+id/blue"

android:title="Blue"

app:showAsAction="never" />

</menu>

colors.xml

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

<resources>

<color name="red">#FF0000</color>

<color name="green">#00FF00</color>

<color name="blue">#0000FF</color>

</resources>

This code creates a menu-driven application that provides a facility to set the background color of the activity. The user can select the background color from the options menu.

Note: This code uses the onCreateOptionsMenu method to inflate the options menu and the onOptionsItemSelected method to handle the menu item clicks. The background color is set using the setBackgroundColor method.

1. **Handler and Threads :**

MAD - Practical - 05 :

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

Here is the full Kotlin code for the application:

MainActivity.kt

import android.os.Bundle

import android.os.Handler

import android.widget.TextView

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

private lateinit var textView: TextView

private var count = 0

private var handler = Handler()

private var runnable: Runnable? = null

private var isRunning = false

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

textView = findViewById(R.id.textView)

startIncrementing()

}

private fun startIncrementing() {

isRunning = true

runnable = object : Runnable {

override fun run() {

count++

textView.text = count.toString()

handler.postDelayed(this, 1000) // increment every 1 second

}

}

handler.post(runnable)

}

fun stopIncrementing(view: android.view.View) {

isRunning = false

handler.removeCallbacks(runnable)

}

}

1. **Asynchronous task :**

MAD - Practical - 06

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

Here is the full Kotlin code for the application:

1) MainActivity.kt

import android.os.Bundle

import android.os.Handler

import android.widget.Button

import android.widget.ProgressBar

import android.widget.TextView

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

private lateinit var progressBar: ProgressBar

private lateinit var progressTextView: TextView

private lateinit var button: Button

private var progress = 0

private var handler = Handler()

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

progressBar = findViewById(R.id.progressBar)

progressTextView = findViewById(R.id.progressTextView)

button = findViewById(R.id.button)

button.setOnClickListener {

startProgress()

}

}

private fun startProgress() {

progress = 0

progressBar.progress = progress

progressTextView.text = "$progress%"

handler.postDelayed(object : Runnable {

override fun run() {

if (progress < 100) {

progress++

progressBar.progress = progress

progressTextView.text = "$progress%"

handler.postDelayed(this, 100) // increment every 100ms

} else {

Toast.makeText(this@MainActivity, "Progress complete!", Toast.LENGTH\_SHORT).show()

}

}

}, 100)

}

}

1. **Android Service :**

MAD -> Practical - 7

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

1) string.xml file :-

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

<resources>

<string name="app\_name">Services\_In\_Android</string>

<string name="heading">Services In Android</string>

<string name="startButtonText">Start the Service</string>

<string name="stopButtonText">Stop the Service</string>

</resources>

2) activity\_main.xml file :-

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

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

<androidx.constraintlayout.widget.ConstraintLayout

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

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#168BC34A"

tools:context=".MainActivity">

<LinearLayout

android:id="@+id/linearLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:orientation="vertical"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="1.0"

tools:ignore="MissingConstraints">

<TextView

android:id="@+id/textView1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="170dp"

android:fontFamily="@font/roboto"

android:text="@string/heading"

android:textAlignment="center"

android:textAppearance="@style/TextAppearance.AppCompat.Large"

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

android:textSize="36sp"

android:textStyle="bold" />

<Button

android:id="@+id/startButton"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginStart="20dp"

android:layout\_marginTop="10dp"

android:layout\_marginEnd="20dp"

android:layout\_marginBottom="20dp"

android:background="#4CAF50"

android:fontFamily="@font/roboto"

android:text="@string/startButtonText"

android:textAlignment="center"

android:textAppearance="@style/TextAppearance.AppCompat.Display1"

android:textColor="#FFFFFF"

android:textStyle="bold" />

<Button

android:id="@+id/stopButton"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginStart="20dp"

android:layout\_marginTop="10dp"

android:layout\_marginEnd="20dp"

android:layout\_marginBottom="20dp"

android:background="#4CAF50"

android:fontFamily="@font/roboto"

android:text="@string/stopButtonText"

android:textAlignment="center"

android:textAppearance="@style/TextAppearance.AppCompat.Display1"

android:textColor="#FFFFFF"

android:textStyle="bold" />

<ImageView

android:id="@+id/imageView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="80dp"

app:srcCompat="@drawable/banner" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

3) create custom service class

import android.app.Service

import android.content.Intent

import android.media.MediaPlayer

import android.os.IBinder

import android.provider.Settings

class NewService : Service() {

// declaring object of MediaPlayer

private lateinit var player:MediaPlayer

// execution of service will start

// on calling this method

override fun onStartCommand(intent: Intent, flags: Int, startId: Int): Int {

// creating a media player which

// will play the audio of Default

// ringtone in android device

player = MediaPlayer.create(this, Settings.System.DEFAULT\_RINGTONE\_URI)

// providing the boolean

// value as true to play

// the audio on loop

player.setLooping(true)

// starting the process

player.start()

// returns the status

// of the program

return START\_STICKY

}

// execution of the service will

// stop on calling this method

override fun onDestroy() {

super.onDestroy()

// stopping the process

player.stop()

}

override fun onBind(intent: Intent): IBinder? {

return null

}

}

4) work with main activity file :-

import android.content.Intent

import android.os.Bundle

import android.view.View

import android.widget.Button

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity(), View.OnClickListener {

// declaring objects of Button class

private var start: Button? = null

private var stop: Button? = null

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

// assigning ID of startButton

// to the object start

start = findViewById<View>(R.id.startButton) as Button

// assigning ID of stopButton

// to the object stop

stop = findViewById<View>(R.id.stopButton) as Button

// declaring listeners for the

// buttons to make them respond

// correctly according to the process

start!!.setOnClickListener(this)

stop!!.setOnClickListener(this)

}

override fun onClick(view: View) {

// process to be performed

// if start button is clicked

if (view === start) {

// starting the service

startService(Intent(this, NewService::class.java))

}

// process to be performed

// if stop button is clicked

else if (view === stop) {

// stopping the service

stopService(Intent(this, NewService::class.java))

}

}

}

5) AndroidManifest.xml file:-

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

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

package="com.example.services\_in\_android">

<application

android:allowBackup="true"

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

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

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

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity android:name=".MainActivity">

<intent-filter>

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

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

</intent-filter>

</activity>

<meta-data

android:name="preloaded\_fonts"

android:resource="@array/preloaded\_fonts" />

<!-- Mention the service name here -->

<service android:name=".NewService"/>

</application>

</manifest>

1. **Notifications :**

MAD - Practical - 08 :- Notification

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

Here is the full Kotlin code for the application:

1) PrimeNumberService.kt

import android.app.Service

import android.content.Intent

import android.os.IBinder

import android.util.Log

class PrimeNumberService : Service() {

override fun onBind(intent: Intent): IBinder? {

return null

}

override fun onStartCommand(intent: Intent, flags: Int, startId: Int): Int {

val lowerBound = intent.getIntExtra("lower\_bound", 0)

val upperBound = intent.getIntExtra("upper\_bound", 0)

Thread {

val primeNumbers = findPrimeNumbers(lowerBound, upperBound)

sendNotification(primeNumbers)

}.start()

return super.onStartCommand(intent, flags, startId)

}

private fun findPrimeNumbers(lowerBound: Int, upperBound: Int): List<Int> {

val primeNumbers = mutableListOf<Int>()

for (i in lowerBound..upperBound) {

if (isPrime(i)) {

primeNumbers.add(i)

}

}

return primeNumbers

}

private fun isPrime(n: Int): Boolean {

if (n <= 1) {

return false

}

for (i in 2..n / 2) {

if (n % i == 0) {

return false

}

}

return true

}

private fun sendNotification(primeNumbers: List<Int>) {

val notificationManager = getSystemService(NOTIFICATION\_SERVICE) as NotificationManager

val notification = NotificationCompat.Builder(this, "prime\_numbers\_channel")

.setContentTitle("Prime Numbers Found")

.setContentText("Prime numbers between range: $primeNumbers")

.setSmallIcon(R.drawable.ic\_notification)

.build()

notificationManager.notify(1, notification)

}

}

2) MainActivity.kt

import android.content.Intent

import android.os.Bundle

import android.widget.Button

import android.widget.EditText

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

findViewById<Button>(R.id.findPrimeNumbersButton).setOnClickListener {

val lowerBound = findViewById<EditText>(R.id.lowerBoundEditText).text.toString().toInt()

val upperBound = findViewById<EditText>(R.id.upperBoundEditText).text.toString().toInt()

val intent = Intent(this, PrimeNumberService::class.java)

intent.putExtra("lower\_bound", lowerBound)

intent.putExtra("upper\_bound", upperBound)

startService(intent)

}

}

}

1. **Read Contacts from Phone :**

MAD - Practical - 09

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

Here is the full Kotlin code for the application:

MainActivity.kt

import android.os.Bundle

import android.provider.ContactsContract

import android.widget.ListView

import androidx.appcompat.app.AppCompatActivity

import androidx.cursoradapter.widget.SimpleCursorAdapter

class MainActivity : AppCompatActivity() {

private lateinit var listView: ListView

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

listView = findViewById(R.id.listView)

val cursor = contentResolver.query(

ContactsContract.Contacts.CONTENT\_URI,

null,

null,

null,

null

)

val adapter = SimpleCursorAdapter(

this,

android.R.layout.simple\_list\_item\_2,

cursor,

arrayOf(

ContactsContract.Contacts.DISPLAY\_NAME,

ContactsContract.Contacts.CONTACT\_STATUS

),

intArrayOf(android.R.id.text1, android.R.id.text2),

0

)

listView.adapter = adapter

}

}

1. **Dynamic Generation of Widgets:**

MAD - Practical - 10

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

Here is the full Kotlin code for the application:

1) MainActivity.kt

import android.content.Intent

import android.os.Bundle

import android.widget.Button

import android.widget.EditText

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

private lateinit var editText: EditText

private lateinit var button: Button

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

editText = findViewById(R.id.editText)

button = findViewById(R.id.button)

button.setOnClickListener {

val number = editText.text.toString().toInt()

val intent = Intent(this, ListActivity::class.java)

intent.putExtra("number", number)

startActivity(intent)

}

}

}

2) ListActivity.kt

import android.os.Bundle

import android.widget.ListView

import androidx.appcompat.app.AppCompatActivity

class ListActivity : AppCompatActivity() {

private lateinit var listView: ListView

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_list)

listView = findViewById(R.id.listView)

val number = intent.getIntExtra("number", 0)

val list = mutableListOf<String>()

for (i in 1..number) {

list.add("Item $i")

}

val adapter = android.widget.ArrayAdapter(this, android.R.layout.simple\_list\_item\_1, list)

listView.adapter = adapter

}

}

1. **Telephony API (Calling):**

MAD -> Practical - 11

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

1) Main Activity.java file :-

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

package com.geeksforgeeks.gfg.dial;

// importing packages

import android.content.Intent;

import android.net.Uri;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

// Binding MainActivity.java with

// activity\_main.xml file

setContentView(R.layout.activity\_main);

}

// This function is called when button is clicked.

public void Call(View v)

{

// Find the EditText by its unique ID

EditText e = (EditText)findViewById(R.id.editText);

// show() method display the toast with message

// "clicked"

Toast.makeText(this, "clicked", Toast.LENGTH\_LONG)

.show();

// Use format with "tel:" and phoneNumber created is

// stored in u.

Uri u = Uri.parse("tel:" + e.getText().toString());

// Create the intent and set the data for the

// intent as the phone number.

Intent i = new Intent(Intent.ACTION\_DIAL, u);

try

{

// Launch the Phone app's dialer with a phone

// number to dial a call.

startActivity(i);

}

catch (SecurityException s)

{

// show() method display the toast with

// exception message.

Toast.makeText(this, "An error occurred", Toast.LENGTH\_LONG)

.show();

}

}

}

2) activity\_main.xml file :-

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

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

<android.support.constraint.ConstraintLayout

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

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

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

<!-- covers entire width of the screen -->

android:layout\_width="match\_parent"

<!-- covers entire height of the screen -->

android:layout\_height="match\_parent"

tools:context="com.example.hp.dial.MainActivity">

<EditText

android:id="@+id/editText"

<!-- covers as much width as required. -->

android:layout\_width="wrap\_content"

<!-- covers as much height as required. -->

android:layout\_height="wrap\_content"

<!-- left spacing from the parent layout-->

android:layout\_marginLeft="8dp"

<!-- right spacing from the parent layout-->

android:layout\_marginRight="8dp"

<!-- top spacing from the parent layout-->

android:layout\_marginTop="65dp"

<!-- hint works as a place holder -->

android:hint="Phone No."

<!-- Expressing the given input should be phone no -->

android:inputType="phone"

app:layout\_constraintHorizontal\_bias="0.503"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<Button

android:id="@+id/button"

<!-- covers as much width as required. -->

android:layout\_width="wrap\_content"

<!-- covers as much height as required. -->

android:layout\_height="wrap\_content"

android:layout\_marginLeft="8dp"

android:layout\_marginRight="8dp"

android:layout\_marginTop="67dp"

<!-- name of function is Call, and it is -->

<!-- invoked when the button is clicked.-->

android:onClick="Call"

android:text="DIAL"

<!-- below are the positions of the button -->

<!-- with respect to editText and parent layout. -->

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/editText"

/>

</android.support.constraint.ConstraintLayout>

1. **Messaging:**

MAD -> Practical - 12

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

1) AndroidManifest.xml file :-

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

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

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

xmlns:dist="http://schemas.android.com/apk/distribution"

package="com.example.gfg">

<uses-permission android:name="android.permission.SEND\_SMS"/>

<dist:module dist:instant="true" />

<application

android:allowBackup="true"

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

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

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

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity android:name=".MainActivity">

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

</manifest>

2) activity\_main . xml :-

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

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

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

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

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

android:layout\_width="match\_parent"

android:orientation="vertical"

android:layout\_marginTop="140dp"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<EditText

android:id="@+id/editText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:ems="10"

android:hint="Enter number"

android:inputType="textPersonName" />

<EditText

android:id="@+id/editText2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:ems="10"

android:hint="Enter message"

android:inputType="textPersonName" />

<Button

android:id="@+id/button"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp"

android:layout\_marginLeft="60dp"

android:layout\_marginRight="60dp"

android:text="SEND" />

</LinearLayout>

3) MainActivity.java :-

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

package com.example.gfg;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.telephony.SmsManager;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

EditText phonenumber,message;

Button send;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

send=findViewById(R.id.button);

phonenumber=findViewById(R.id.editText);

message=findViewById(R.id.editText2);

send.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {

String number=phonenumber.getText().toString();

String msg=message.getText().toString();

try {

SmsManager smsManager=SmsManager.getDefault();

smsManager.sendTextMessage(number,null,msg,null,null);

Toast.makeText(getApplicationContext(),"Message Sent",Toast.LENGTH\_LONG).show();

}catch (Exception e)

{

Toast.makeText(getApplicationContext(),"Some fields is Empty",Toast.LENGTH\_LONG).show();

}

}

});

}

}

1. **File I/O :**

MAD -> Practical - 13

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

1) activity\_main.xml file:-

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

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

<RelativeLayout

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"

tools:context=".MainActivity">

<!--text view for heading-->

<TextView

android:id="@+id/idTVHeader"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="30dp"

android:gravity="center\_horizontal"

android:padding="5dp"

android:text="Welcome to Geeks for Geeks \n Login Form"

android:textAlignment="center"

android:textColor="@color/purple\_700"

android:textSize="18sp" />

<!--edit text for user name-->

<EditText

android:id="@+id/idEdtUserName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/idTVHeader"

android:layout\_marginStart="10dp"

android:layout\_marginTop="50dp"

android:layout\_marginEnd="10dp"

android:hint="Enter UserName"

android:inputType="textEmailAddress" />

<!--edit text for user password-->

<EditText

android:id="@+id/idEdtPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/idEdtUserName"

android:layout\_marginStart="10dp"

android:layout\_marginTop="20dp"

android:layout\_marginEnd="10dp"

android:hint="Enter Password"

android:inputType="textPassword" />

<!--button to register our new user-->

<Button

android:id="@+id/idBtnLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/idEdtPassword"

android:layout\_marginStart="10dp"

android:layout\_marginTop="20dp"

android:layout\_marginEnd="10dp"

android:text="Login"

android:textAllCaps="false" />

</RelativeLayout>

2) activity\_home.xml file:-

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

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

<RelativeLayout

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"

tools:context=".HomeActivity">

<!--text view for displaying heading-->

<TextView

android:id="@+id/idTVHeader"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:gravity="center\_horizontal"

android:text="Welcome back again to Geeks for Geeks"

android:textAlignment="center"

android:textColor="@color/purple\_700"

android:textSize="18sp" />

<!--text view for displaying user name-->

<TextView

android:id="@+id/idTVUserName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/idTVHeader"

android:layout\_centerInParent="true"

android:layout\_marginTop="20dp"

android:gravity="center\_horizontal"

android:text="UserName"

android:textAlignment="center"

android:textColor="@color/purple\_700"

android:textSize="25sp" />

<!--button for making user log out-->

<Button

android:id="@+id/idBtnLogout"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/idTVUserName"

android:layout\_centerInParent="true"

android:layout\_marginTop="10dp"

android:text="LogOut"

android:textAllCaps="false" />

</RelativeLayout>

3) HomeActivity.java file :-

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

import android.content.Intent;

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.parse.ParseUser;

public class HomeActivity extends AppCompatActivity {

// creating a variable

// for our text view..

private TextView userNameTV;

// button for logout

private Button logoutBtn;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_home);

logoutBtn = findViewById(R.id.idBtnLogout);

// initializing our variables

userNameTV = findViewById(R.id.idTVUserName);

// getting data from intent.

String name = getIntent().getStringExtra("username");

// setting data to our text view.

userNameTV.setText(name);

// initializing click listener for logout button

logoutBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// calling a method to logout our user.

ParseUser.logOutInBackground(e -> {

if (e == null) {

Toast.makeText(HomeActivity.this, "User Logged Out", Toast.LENGTH\_SHORT).show();

Intent i = new Intent(HomeActivity.this, MainActivity.class);

startActivity(i);

finish();

}

});

}

});

}

}

4) MainActivity.java file :-

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

import android.content.Intent;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.parse.ParseUser;

public class MainActivity extends AppCompatActivity {

// creating variables for our edit text and buttons.

private EditText userNameEdt, passwordEdt;

private Button loginBtn;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

// initializing our edit text and buttons.

userNameEdt = findViewById(R.id.idEdtUserName);

passwordEdt = findViewById(R.id.idEdtPassword);

loginBtn = findViewById(R.id.idBtnLogin);

// adding on click listener for our button.

loginBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// on below line we are getting data from our edit text.

String userName = userNameEdt.getText().toString();

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

// checking if the entered text is empty or not.

if (TextUtils.isEmpty(userName) && TextUtils.isEmpty(password)) {

Toast.makeText(MainActivity.this, "Please enter user name and password", Toast.LENGTH\_SHORT).show();

}

// calling a method to login our user.

loginUser(userName, password);

}

});

}

private void loginUser(String userName, String password) {

// calling a method to login a user.

ParseUser.logInInBackground(userName, password, (parseUser, e) -> {

// after login checking if the user is null or not.

if (parseUser != null) {

// if the user is not null then we will display a toast message

// with user login and passing that user to new activity.

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

Intent i = new Intent(MainActivity.this, HomeActivity.class);

i.putExtra("username", userName);

startActivity(i);

} else {

// display a toast message when user logout of the app.

ParseUser.logOut();

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

}

});

}

1. **Shared Preferences :**

MAD - Practical - 14

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

Here is the full Kotlin code for the application:

1) MainActivity.kt

import android.os.Bundle

import android.view.View

import android.widget.Button

import androidx.appcompat.app.AppCompatActivity

import androidx.core.content.ContextCompat

class MainActivity : AppCompatActivity() {

private lateinit var button: Button

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

button = findViewById(R.id.button)

val sharedPreferences = getSharedPreferences("background\_color", MODE\_PRIVATE)

val selectedColor = sharedPreferences.getInt("color", ContextCompat.getColor(this, R.color.default\_color))

window.decorView.setBackgroundColor(selectedColor)

button.setOnClickListener {

ColorPickerDialog(this) { color ->

window.decorView.setBackgroundColor(color)

sharedPreferences.edit().putInt("color", color).apply()

}.show()

}

}

}

2) ColorPickerDialog.kt

import android.app.Dialog

import android.content.Context

import android.graphics.Color

import android.view.View

import android.widget.Button

class ColorPickerDialog(context: Context, private val onColorSelected: (Int) -> Unit) : Dialog(context) {

init {

setContentView(R.layout.dialog\_color\_picker)

findViewById<Button>(R.id.button\_red).setOnClickListener {

onColorSelected(Color.RED)

dismiss()

}

findViewById<Button>(R.id.button\_green).setOnClickListener {

onColorSelected(Color.GREEN)

dismiss()

}

findViewById<Button>(R.id.button\_blue).setOnClickListener {

onColorSelected(Color.BLUE)

dismiss()

}

}

}

1. **Database CRUD Operations :**

MAD - Practical - 15

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

Here is the full Kotlin code for the application:

1) User.kt

data class User(

val id: Int,

val name: String,

val address: String,

val contactNumber: String

)

2) UserDatabase.kt

import android.content.Context

import android.database.sqlite.SQLiteDatabase

import android.database.sqlite.SQLiteOpenHelper

class UserDatabase(context: Context) :

SQLiteOpenHelper(context, "user\_database", null, 1) {

override fun onCreate(db: SQLiteDatabase?) {

db?.execSQL("CREATE TABLE users (id INTEGER PRIMARY KEY, name TEXT, address TEXT, contact\_number TEXT)")

}

override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {

db?.execSQL("DROP TABLE IF EXISTS users")

onCreate(db)

}

}

3) UserDao.kt

import android.content.Context

import android.database.sqlite.SQLiteDatabase

class UserDao(context: Context) {

private val database: SQLiteDatabase

init {

database = UserDatabase(context).writableDatabase

}

fun addUser(user: User) {

database.execSQL("INSERT INTO users VALUES (?, ?, ?, ?)",

arrayOf((link unavailable), user.name, user.address, user.contactNumber))

}

fun updateUser(user: User) {

database.execSQL("UPDATE users SET name = ?, address = ?, contact\_number = ? WHERE id = ?",

arrayOf(user.name, user.address, user.contactNumber, (link unavailable)))

}

fun deleteUser(id: Int) {

database.execSQL("DELETE FROM users WHERE id = ?", arrayOf(id))

}

fun getAllUsers(): List<User> {

val cursor = database.rawQuery("SELECT \* FROM users", null)

val users = mutableListOf<User>()

while (cursor.moveToNext()) {

users.add(User(

cursor.getInt(0),

cursor.getString(1),

cursor.getString(2),

cursor.getString(3)

))

}

cursor.close()

return users

}

}

4) MainActivity.kt

import android.os.Bundle

import android.widget.Button

import android.widget.EditText

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

private lateinit var userDao: UserDao

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

userDao = UserDao(this)

findViewById<Button>(R.id.addButton).setOnClickListener {

val id = findViewById<EditText>(R.id.idEditText).text.toString().toInt()

val name = findViewById<EditText>(R.id.nameEditText).text.toString()

val address = findViewById<EditText>(R.id.addressEditText).text.toString()

val contactNumber = findViewById<EditText>(R.id.contactNumberEditText).text.toString()

userDao.addUser(User(id, name, address, contactNumber))

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

}

findViewById<Button>(R.id.updateButton).setOnClickListener {

val id = findViewById<EditText>(R.id.idEditText).text.toString().toInt()

val name = findViewById<EditText>(R.id.nameEditText).text.toString()

val address = findViewById<EditText>(R.id.addressEditText).text.toString()

val contactNumber = findViewById<EditText>(R.id.contactNumberEditText).text.toString()

userDao.updateUser(User(id, name, address, contactNumber))

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

}

findViewById<Button>(R.id.deleteButton).setOnClickListener {

val id = findViewById<EditText>(R.id.idEditText).text.toString().toInt()

userDao.deleteUser(id)

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

}

findViewById<Button>(R.id.listButton).setOnClickListener {

val users = userDao.getAllUsers()

Toast.makeText(this, users.toString(), Toast.LENGTH\_SHORT).show()

}

}

}

1. **Data Driven Authentication :**

MAD - Practical - 16 : Data Driven Authentication

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

Here is the full Kotlin code for the application:

1) User.kt

data class User(

val id: String,

val password: String

)

2) UserDatabase.kt

import android.content.Context

import android.database.sqlite.SQLiteDatabase

import android.database.sqlite.SQLiteOpenHelper

class UserDatabase(context: Context) :

SQLiteOpenHelper(context, "user\_database", null, 1) {

override fun onCreate(db: SQLiteDatabase?) {

db?.execSQL("CREATE TABLE users (id TEXT PRIMARY KEY, password TEXT)")

}

override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {

db?.execSQL("DROP TABLE IF EXISTS users")

onCreate(db)

}

}

3) UserDao.kt :-

import android.content.Context

import android.database.sqlite.SQLiteDatabase

class UserDao(context: Context) {

private val database: SQLiteDatabase

init {

database = UserDatabase(context).writableDatabase

}

fun addUser(user: User) {

database.execSQL("INSERT INTO users VALUES (?, ?)",

arrayOf((link unavailable), user.password))

}

fun authenticateUser(id: String, password: String): Boolean {

val cursor = database.rawQuery("SELECT \* FROM users WHERE id = ? AND password = ?",

arrayOf(id, password))

val isAuthenticated = cursor.count > 0

cursor.close()

return isAuthenticated

}

}

4) MainActivity.kt :-

import android.os.Bundle

import android.widget.Button

import android.widget.EditText

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

private lateinit var userDao: UserDao

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

userDao = UserDao(this)

findViewById<Button>(R.id.loginButton).setOnClickListener {

val id = findViewById<EditText>(R.id.idEditText).text.toString()

val password = findViewById<EditText>(R.id.passwordEditText).text.toString()

if (userDao.authenticateUser(id, password)) {

Toast.makeText(this, "Login successful", Toast.LENGTH\_SHORT).show()

} else {

Toast.makeText(this, "Invalid id or password", Toast.LENGTH\_SHORT).show()

}

}

}

}

1. **Animation :**

MAD - Practical - 17

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

Here is the full Kotlin code for the application:

MainActivity.kt

import android.os.Bundle

import android.view.animation.Animation

import android.view.animation.AnimationUtils

import android.widget.ImageView

import android.widget.Spinner

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

private lateinit var spinner: Spinner

private lateinit var imageView: ImageView

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

spinner = findViewById(R.id.spinner)

imageView = findViewById(R.id.imageView)

spinner.setOnItemSelectedListener { \_, \_, position, \_ ->

val animationName = spinner.getItemAtPosition(position).toString()

val animation = when (animationName) {

"Fade In" -> AnimationUtils.loadAnimation(this, android.R.anim.fade\_in)

"Fade Out" -> AnimationUtils.loadAnimation(this, android.R.anim.fade\_out)

"Rotate" -> AnimationUtils.loadAnimation(this, android.R.anim.rotate)

"Scale" -> AnimationUtils.loadAnimation(this, android.R.anim.scale)

else -> null

}

animation?.let { imageView.startAnimation(it) }

}

}

}

activity\_main.xml

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

<LinearLayout xmlns:android="(link unavailable)"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<Spinner

android:id="@+id/spinner"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:entries="@array/animations" />

<ImageView

android:id="@+id/imageView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:src="@mipmap/ic\_launcher" />

</LinearLayout>

strings.xml

<resources>

<string-array name="animations">

<item>Fade In</item>

<item>Fade Out</item>

<item>Rotate</item>

<item>Scale</item>

</string-array>

</resources>

This code creates an application that contains a spinner and an image. The spinner contains names of animations. Upon selecting an animation name, the selected animation is applied on the image view.

Note: This code uses the AnimationUtils class to load the animations and the startAnimation method to apply the animation to the image view.

1. **Audio Player :**

MAD -> Practical - 18

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

1) Activity\_main.xml file :-

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

<?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/activity\_main"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/colorPrimary"

android:orientation="vertical"

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

tools:context=".MainActivity">

<ImageView

android:id="@+id/imageView"

android:layout\_width="match\_parent"

android:layout\_height="430dp"

android:background="@drawable/download"

android:contentDescription="@string/todo" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_margin="20dp"

android:background="@color/colorAccent"

android:orientation="horizontal"

android:padding="10dp">

<Button

android:id="@+id/pause"

style="@style/Widget.AppCompat.Button.Borderless.Colored"

android:layout\_width="125dp"

android:layout\_height="match\_parent"

android:background="@android:drawable/ic\_media\_pause"

android:onClick="musicpause" />

<Button

android:id="@+id/start"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="125dp"

android:layout\_height="match\_parent"

android:background="@android:drawable/ic\_media\_play"

android:onClick="musicplay" />

<Button

android:id="@+id/stop"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="125dp"

android:layout\_height="match\_parent"

android:background="@android:drawable/ic\_delete"

android:onClick="musicstop" />

</LinearLayout>

</LinearLayout>

2) Main Activity.java file :-

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

package com.example.amusinz;

import android.media.MediaPlayer;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

public class MainActivity

extends AppCompatActivity {

// Instantiating the MediaPlayer class

MediaPlayer music;

@Override

protected void onCreate(

Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

// Adding the music file to our

// newly created object music

music = MediaPlayer.create(

this, R.raw.sound);

}

// Playing the music

public void musicplay(View v)

{

music.start();

}

// Pausing the music

public void musicpause(View v)

{

music.pause();

}

// Stopping the music

public void musicstop(View v)

{

music.stop();

music

= MediaPlayer.create(

this, R.raw.sound);

}

}

1. **Web Browser :**

MAD - Practical - 19 Web Browser :

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

Here is the full Kotlin code for the application:

1) MainActivity.kt

import android.os.Bundle

import android.view.View

import android.webkit.WebView

import android.webkit.WebViewClient

import android.widget.Button

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

private lateinit var webView: WebView

private lateinit var backButton: Button

private lateinit var forwardButton: Button

private lateinit var clearButton: Button

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

webView = findViewById(R.id.webView)

backButton = findViewById(R.id.backButton)

forwardButton = findViewById(R.id.forwardButton)

clearButton = findViewById(R.id.clearButton)

webView.webViewClient = object : WebViewClient() {

override fun shouldOverrideUrlLoading(view: WebView, url: String): Boolean {

view.loadUrl(url)

return true

}

}

backButton.setOnClickListener {

if (webView.canGoBack()) {

webView.goBack()

}

}

forwardButton.setOnClickListener {

if (webView.canGoForward()) {

webView.goForward()

}

}

clearButton.setOnClickListener {

webView.clearHistory()

}

webView.loadUrl("(link unavailable)")

}

}

1. **Location Based Services :**

MAD - Practical - 20

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

Here is the full Kotlin code for the application:

MainActivity.kt

import android.Manifest

import android.content.Context

import android.content.pm.PackageManager

import android.location.Location

import android.location.LocationManager

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.widget.TextView

import androidx.core.app.ActivityCompat

class MainActivity : AppCompatActivity() {

private lateinit var locationManager: LocationManager

private lateinit var latitudeTextView: TextView

private lateinit var longitudeTextView: TextView

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

locationManager = getSystemService(Context.LOCATION\_SERVICE) as LocationManager

latitudeTextView = findViewById(R.id.latitudeTextView)

longitudeTextView = findViewById(R.id.longitudeTextView)

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,

arrayOf(Manifest.permission.ACCESS\_FINE\_LOCATION),

1

)

} else {

val location = locationManager.getLastKnownLocation(LocationManager.GPS\_PROVIDER)

updateLocation(location)

}

}

override fun onRequestPermissionsResult(

requestCode: Int,

permissions: Array<out String>,

grantResults: IntArray

) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults)

if (requestCode == 1 && grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

val location = locationManager.getLastKnownLocation(LocationManager.GPS\_PROVIDER)

updateLocation(location)

}

}

private fun updateLocation(location: Location?) {

location?.let {

latitudeTextView.text = "Latitude: ${it.latitude}"

longitudeTextView.text = "Longitude: ${it.longitude}"

}

}

}

activity\_main.xml

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

<LinearLayout xmlns:android="(link unavailable)"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:id="@+id/latitudeTextView"

android:layout\_width="match\_parent"

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

<TextView

android:id="@+id/longitudeTextView"

android:layout\_width="match\_parent"

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

</LinearLayout>

This code creates an application that keeps track of the location (coordinates) of the device and displays the values of longitude and latitude on the screen.

Note: This code requires the ACCESS\_FINE\_LOCATION permission in the AndroidManifest.xml file:

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

Also, this code uses the LocationManager class to get the last known location of the device.