Write an android code to turn ON /OFF the Bluetooth.

- AndroidManifest.xml:

<manifest ...>

<!-- Add Bluetooth permissions -->

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

<uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" />

<application ...>

...

</application>

</manifest>

- MainActivity.java:

import android.bluetooth.BluetoothAdapter;

import android.content.Intent;

import android.os.Bundle;

import android.widget.CompoundButton;

import android.widget.Toast;

import android.widget.ToggleButton;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

private BluetoothAdapter bluetoothAdapter;

private ToggleButton toggleButton;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

toggleButton = findViewById(R.id.toggleButton);

if (bluetoothAdapter == null) {

// Device doesn't support Bluetooth

Toast.makeText(this, "Bluetooth is not supported on this device", Toast.LENGTH\_SHORT).show();

finish(); // Close the app

return;

}

toggleButton.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {

@Override

public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {

if (isChecked) {

// Turn ON Bluetooth

enableBluetooth();

} else {

// Turn OFF Bluetooth

disableBluetooth();

}

}

});

// Set initial state of the toggle button based on Bluetooth status

boolean isEnabled = bluetoothAdapter.isEnabled();

toggleButton.setChecked(isEnabled);

}

private void enableBluetooth() {

if (!bluetoothAdapter.isEnabled()) {

Intent enableBluetoothIntent = new Intent(BluetoothAdapter.ACTION\_REQUEST\_ENABLE);

startActivityForResult(enableBluetoothIntent, 1);

}

}

private void disableBluetooth() {

if (bluetoothAdapter.isEnabled()) {

bluetoothAdapter.disable();

}

}

}

activity\_main.xml:

<!-- activity\_main.xml -->

<RelativeLayout ...>

<ToggleButton

android:id="@+id/toggleButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textOff="Bluetooth OFF"

android:textOn="Bluetooth ON"

android:layout\_centerInParent="true" />

</RelativeLayout>

Creating an Android "Hello World" Application with PhoneGap.

Prerequisites:

- Install Node.js: Make sure you have Node.js installed on your system. You can download it from Node.js website.

- Install PhoneGap/Cordova: Install PhoneGap globally using npm (Node Package Manager). Open your terminal or command prompt and run:

bash

npm install -g phonegap@latest

- Java Development Kit (JDK): Install JDK and set the Java environment variables.

- Android Studio: Install Android Studio and set up the necessary Android SDKs and tools.

- Creating a PhoneGap Project:

Open a terminal or command prompt and navigate to the directory where you want to create the project.

Run the following command to create a new PhoneGap project:

- phonegap create hello-world com.example.helloworld HelloWorld

This command creates a new PhoneGap project named "hello-world" with the package name com.example.helloworld and the app name "HelloWorld".

- Navigate to the project directory:

bash

cd hello-world

- Add Android platform to the project:

bash

phonegap platform add android

- Edit the HTML/JavaScript:

- Navigate to the www folder inside your project directory. Open the index.html file in a text editor and modify it to display "Hello World".

Here's an example:

html

Copy code

<!DOCTYPE html>

<html>

<head>

<title>Hello World</title>

<script type="text/javascript" src="cordova.js"></script>

<script type="text/javascript">

document.addEventListener("deviceready", onDeviceReady, false);

function onDeviceReady() {

// Handle the device ready event

document.getElementById("hello").innerHTML = "Hello, PhoneGap!";

}

</script>

</head>

<body>

<h1 id="hello">Waiting for device ready event...</h1>

</body>

</html>

- Build the Android app:

- Run the following command to build the Android app:

bash

phonegap build android

Run the app:

Connect your Android device to your computer or use an emulator.

- Run the app on the connected Android device/emulator using:

bash

phonegap run android

This will install the "Hello World" app on your Android device/emulator. You should see the "Hello, PhoneGap!" message displayed when the app starts.

Remember to replace the com.example.helloworld with your desired package name, and modify the HTML/JavaScript in index.html to customize the app further.

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

.Create an Android Application that accept multiple items from one activity and pass to

next activity and display the calculation of the first and second activity data on third.

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

private EditText item1EditText, item2EditText;

private Button calculateButton;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

item1EditText = findViewById(R.id.editTextItem1);

item2EditText = findViewById(R.id.editTextItem2);

calculateButton = findViewById(R.id.calculateButton);

calculateButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String item1 = item1EditText.getText().toString();

String item2 = item2EditText.getText().toString();

Intent intent = new Intent(MainActivity.this, SecondActivity.class);

intent.putExtra("item1", item1);

intent.putExtra("item2", item2);

startActivity(intent);

}

});

}

}