**INTERNSHIP**



DEPARTMENT OF COMPUTER ENGINEERING

GOVERNMENT POLYTECHNIQ FOR GIRLS

TOPIC :- ANDROID

NAME :- GOHIL MITANSHI HITESHBHAI

GUIDE BY:- Mihir Parmar

ENROLLMENT NO :- 226150307063

**Student Details :-**

NAME :- GOHIL MITANSHI HITESHBHAI

ENROLLMENT NO :- 226150307063

DEPARTMENT OF COMPUTER ENGINEERING

**Company Details :-**

GUIDE BY:- Mihir Parmar

NAME:-Leaf Drive

ADDRESS:-225,Mahek Icon,sumul Diary Rd,surat,gujarat 395008

EMAIL:-www.ligerinfotech.com

PHONE:-7405212557

**INTENSHIP MODE**:-OFFLINE

**TOPIC NAME :-**

ANDROID

DATE :- 10-0-2023 TO 23-08-2023

**INDEX**

Week 1 .

* How to download or install software.
* Android Studio
* Introduction about android studio and its structure
* Xml - working

- Types of layout

- Linear Layout

Week 2 .

- Android Component

- How to link one page to another page

- Start a mini project

- Linear Layout

- Relative Layout

-Basic learn

- Button

Week 3 .

-Making Tic Tac Toe Game

- design

-Logic -Implement

-Add App Logo

-Run the app in Mobile

Week 4 .

* Splash Screen
* One page to another page
* Making Form
* Implement
* Run Application

Week 5 .

* Calculator
* Logic
* Implementation
* Add Icon
* Run Application

Week 6 .

* MyTask
* Design
* Implementation
* Add Icon
* Run Application

**Week 1**

**DAY 1 : How to download or install software.**

* Android Studio

Step 1: Head over to this link to get the Android Studio executable or zip file.

Step 2: Click on the Download Android Studio Button.

Step 3: After the downloading has finished, open the file from downloads and run it. It will prompt the following dialog box.

Step 4: It will start the installation, and once it is completed, it will be like the image shown below.

Step 5: Once “Finish” is clicked, it will ask whether the previous settings need to be imported [if the android studio had been installed earlier], or not.

Step 6: This will start the Android Studio.

Step 7: After it has found the SDK components, it will redirect to the Welcome dialog box. .

Step 8: Now it is time to download the SDK components.

Step 9: Click on Start a new Android Studio project to build a new app.

**DAY 2:Introduction about android studio and its structure.**

**App**

* **Manifests**
* **Java**
* **Resource**

**Manifests :-**

All things are handle like java ,resource etc....

**Java :-**

**-class 1.activity**

**2.function**

**Resource:-**

**-drawable**

**-layout**

**-mipmap**

**-value**

**-xml**

**To make a project:-**

-project name

-pkg name

-folder path

-version

**Intent**

-Implicit intent

-Explicit intent

An Intent is a messaging object you can use to request an action from another app component. Although intents facilitate communication between components in several ways, there are three fundamental use cases

**Intent types:**There are two types of intents:

* **Explicit intents** specify which application will satisfy the intent, by supplying either the target app's package name or a fully-qualified component class name you know the class name of the activity or service you want to start.
* **Implicit intents** do not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it.

**DAY 3: Xml - working - Types of layout**

**What is xml?**

-(extensible markup language) is a flexible way to create common information and share both the format and the data on the world wild web,intranets ,and else where.

**What is layout?**

-A layout defines the structure for a user interface in your app, such as in an [activity](https://developer.android.com/guide/components/activities). All elements in the layout are built using a hierarchy of [View](https://developer.android.com/reference/android/view/View) and [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup) objects.

**Types of layout?**

-Linear layout

-Relative layout

-Constraint layout

-Table layout

-Frame layout

-list view

-grid view

**-Absolute layout**

**What is linear layout?**

LinearLayout is a view group that aligns all children in a single direction, vertically or horizontally. You can specify the layout direction with the android:orientation attribute. Note: For better performance and tooling support, you should instead build your layout with ConstraintLayout.

<view

android:background=”@color/black”

android:layout\_width=”wrap\_content”

android:layout\_height=””/>

<TextView

android:text=”hello”

android:layout\_width=”wrap\_content”

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

**DAY 4: LINEAR LAYOUT**

**What is linear layout?**

LinearLayout is a view group that aligns all children in a single direction, vertically or horizontally. You can specify the layout direction with the android:orientation attribute. Note: For better performance and tooling support, you should instead build your layout with ConstraintLayout.

**<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <View…….**

**/>**

**<view…………**

**/>**

**<View………..**

**/>**

**</linearLayout>**

**DAY 5: Relative layout**

Relative layout is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent RelativeLayout area (such as aligned to the bottom, left or center)

.

**<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">**

**<TextView………..**

**/>**

**<TextView………..**

**/>**

**<TextView………..**

**/>**

**</RelativeLayout**

**Week 2**

**DAY 1: Android component**

-Activity

-Service

-Broadcast Receiver

-Content Provider

Activity**:-** An activity is a class that is considered as an entry point for users that represents a single screen.

Service:- A service is a component that runs in the background, it acts as an invisible worker of our application.

Broadcast Receiver:- Broadcast Receiver is a component that responds to broadcast messages from another application or the same system.

Content Provider:- Content Provider is a component that allows applications to share data among multiple application

**Day 2 : Linear Layout practice**

**<LinearLayout**

**android:layout\_width="match\_parent"**

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

**<View**

**android:layout\_width="90dp"**

**android:layout\_height="100dp"**

**android:background="@color/purple" />**

**<View**

**android:layout\_width="90dp"**

**android:layout\_height="100dp"**

**android:background="@color/orange" />**

**<View**

**android:layout\_width="90dp"**

**android:layout\_height="100dp"**

**android:background="@color/light\_blue" />**

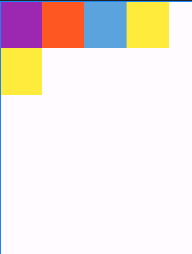
**<View**

**android:layout\_width="90dp"**

**android:layout\_height="100dp"**

**android:background="@color/yello" />**

**</LinearLayout>**

****

**Day 3 : Relative Layout practice**

<RelativeLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:text="hdnvbbuhuyhu"

android:textSize="15dp"

android:textColor="@color/black"/>

<TextView

android:text="hdjh"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:background="@color/yello"

android:id="@+id/date"

android:layout\_below="@+id/name"/>

<!-- android:layout\_toRightOf="@id/name"-->

<View

android:layout\_width="100dp"

android:layout\_height="100dp"

android:background="@color/blue"

android:id="@+id/mitu"

android:layout\_below="@+id/date"/>

<!-- android:layout\_toRightOf="@id/date" />-->

</RelativeLayout>



**Day 4 :How to make Button**

<Button

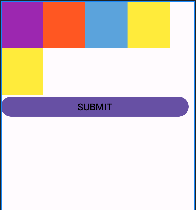
android:layout\_width="400dp"

android:layout\_height="50dp"

android:text="SUBMIT"

android:textColor="@color/black"

android:textSize="20dp"/>



**Day 5 :Basic things Learn**

* **add image**
* **add background color**
* **gravity**
* **text Size**
* **add image**

**Add Image**

<ImageView

android:layout\_width="80dp"

android:layout\_height="80dp"

android:id="@+id/first"

android:layout\_margin="10dp"

android:background="@drawable/wheather"/>

**Gravity**

android:gravity= “center”

**Text Size**

android:textSize="20dp"

**Week 3**

**Day 1 : Tic Tac Toe game**

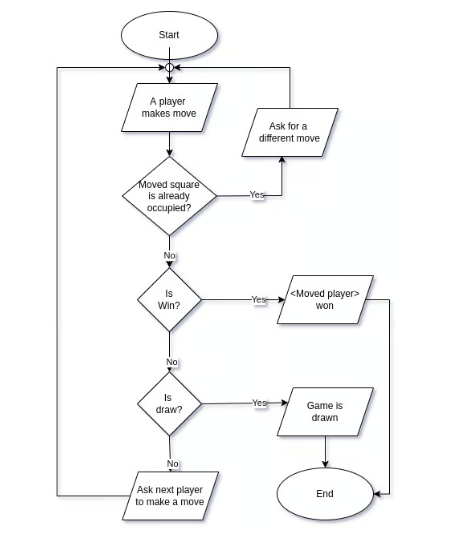
* Design

Tic Tac Toe Design using Gride Layout

The **grid** is composed of a set of infinitely thin lines that separate the viewing area into cells. Throughout the API, grid lines are referenced by grid indices. A grid with N columns has N + 1 grid indices that run from 0 through N inclusive.



**Day 2 : Logic**

****

**Day 3 : Implementation**

package com.example.tictactoe;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

String b1, b2, b3, b4, b5, b6, b7, b8, b9;

Button btn1, btn2, btn3, btn4, btn5, btn6, btn7, btn8, btn9,restart;

int flag = 0;

int count = 0;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

init(); }

private void init() {

btn1 = findViewById(R.id.btn1);

btn2 = findViewById(R.id.btn2);

btn3 = findViewById(R.id.btn3);

btn4 = findViewById(R.id.btn4);

btn5 = findViewById(R.id.btn5);

btn6 = findViewById(R.id.btn6);

btn7 = findViewById(R.id.btn7);

btn8 = findViewById(R.id.btn8);

btn9 = findViewById(R.id.btn9);

// restart = findViewById(R.id.restart);

}

public void check(View view) {

Button btncurrent = (Button) view;

if (btncurrent.getText().toString().equals("")) {

count++;

if (flag == 0) {

btncurrent.setText("X");

flag = 1;

} else {

btncurrent.setText("0");

flag = 0;

} if (count > 4) {

b1 = btn1.getText().toString();

b2 = btn2.getText().toString();

b3 = btn3.getText().toString();

b4 = btn4.getText().toString();

b5 = btn5.getText().toString();

b6 = btn6.getText().toString();

b7 = btn7.getText().toString();

b8 = btn8.getText().toString();

b9 = btn9.getText().toString();

//condition

if (b1.equals(b2) && b2.equals(b3) && !b1.equals("")) {

//1

Toast.makeText(this, "winner is " + b1, Toast.LENGTH\_SHORT).show();

newgame();

} else if (b4.equals(b5) && b5.equals(b6) && !b4.equals("")) {

//2

newgame();

Toast.makeText(this, "winner is " + b7, Toast.LENGTH\_SHORT).show();

} else if (b7.equals(b8) && b8.equals(b9) && !b9.equals("")) {

newgame();

Toast.makeText(this, "winner is " + b7, Toast.LENGTH\_SHORT).show();

//3

} else if (b1.equals(b4) && b4.equals(b7) && !b1.equals("")) {

newgame();

Toast.makeText(this, "winner is " + b1, Toast.LENGTH\_SHORT).show();

//4

} else if (b2.equals(b5) && b5.equals(b8) && !b8.equals("")) {

newgame();

Toast.makeText(this, "winner is " + b2, Toast.LENGTH\_SHORT).show();

//5

} else if (b3.equals(b6) && b6.equals(b9) && !b3.equals("")) {

newgame();

Toast.makeText(this, "winner is " + b3, Toast.LENGTH\_SHORT).show();

//6

} else if (b1.equals(b5) && b5.equals(b9) && !b1.equals("")) {

newgame();

Toast.makeText(this, "winner is " + b1, Toast.LENGTH\_SHORT).show();

//7

} else if (b3.equals(b5) && b5.equals(b7) && !b3.equals("")) {

newgame();

Toast.makeText(this, "winner is " + b3, Toast.LENGTH\_SHORT).show();

//8

}else if(!b1.equals("") && !b2.equals("") && !b3.equals("") && !b4.equals("") &&

!b5.equals("") && !b6.equals("") && !b7.equals("") && !b8.equals("") &&!b9.equals("")){

Toast.makeText(this,"Match is Drawn",Toast.LENGTH\_SHORT).show();

newgame();

} } } }

public void newgame(){

btn1.setText("");

btn2.setText("");

btn3.setText("");

btn4.setText("");

btn5.setText("");

btn6.setText("");

btn7.setText("");

btn8.setText("");

btn9.setText("");

count=0;

flag=0;

}

}

**Day 4 :Add App Logo**

<application

android:allowBackup="true"

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

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

android:icon="@drawable/tic"

android:label="Tic Tac Toe"

android:roundIcon="@drawable/tic"

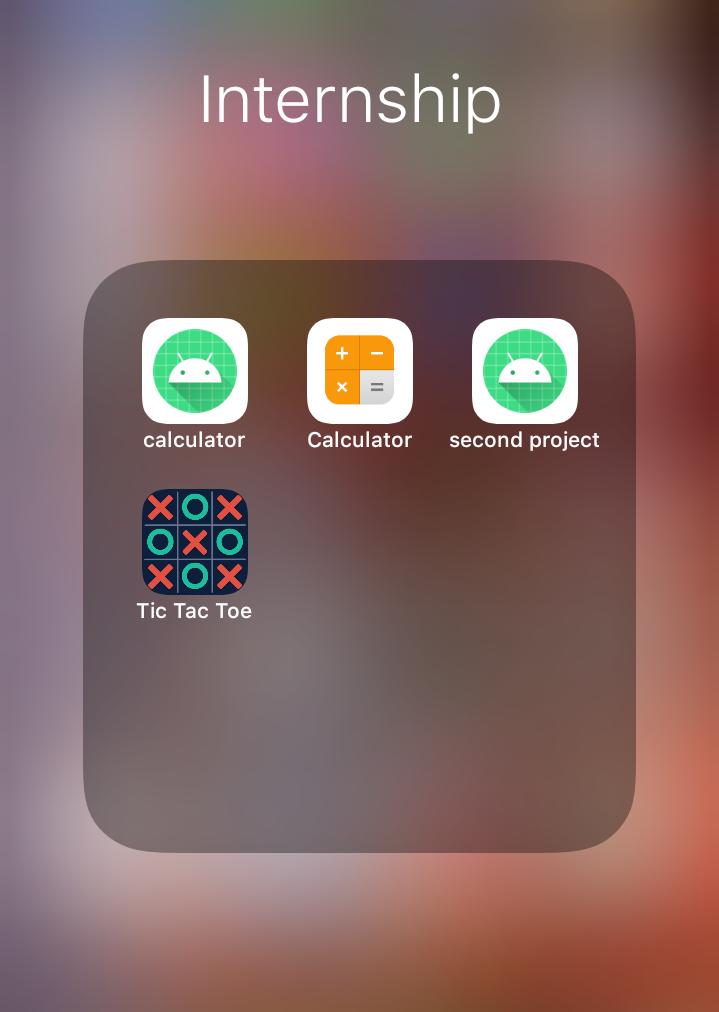
android:supportsRtl="true"

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

tools:targetApi="31">

</application>





**Day 5 :Run Android Project**

****



**Week 4**

**Day 1 :Form Making**

**Splash Screen :**

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_splashscreen);

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

Intent ihome = new Intent(splashscreen.this, secondActivity.class);

startActivity(ihome);

finish();

}

},4000); }

Splash Screen



**DAY 2 :How to link one page to another page**

**Implicit Intent** doesn't specifiy the component. In such case, intent provides information of available components provided by the system that is to be invoked.

For example, you may write the following code to view the webpage.

1. Intent intent=**new** Intent(Intent.ACTION\_VIEW);
2. intent.setData(Uri.parse("http://www.javatpoint.com"));
3. startActivity(intent);

**Explicit Intent** specifies the component. In such case, intent provides the external class to be invoked.

1. Intent i = **new** Intent(getApplicationContext(), ActivityTwo.**class**);
2. startActivity(i);

**Day 3 : Form Making**

**Form**

In this this form I can use : LinearLayout , ScrollView , TextView , ImageView , EditText , Button

<EditText

android:id="@+id/name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

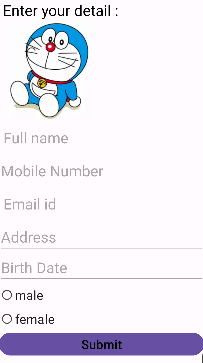
android:layout\_margin="5dp"

android:hint="Full name "

android:inputType="text"

android:textColor="@color/black"

android:textSize="30dp" />

****

**Day 4 : Text Enter**

click.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String name = edit.getText().toString();

String mnumber = mobilenumber.getText().toString();

String emailadd =email.getText().toString();

String addres =addr.getText().toString();

String date=birth.getText().toString();

Intent iintent =new Intent(secondActivity.this,MainActivity2.class);

iintent.putExtra("key",name);

iintent.putExtra("key1",mnumber);

iintent.putExtra("key2",emailadd);

iintent.putExtra("key3",addres);

iintent.putExtra("key4",date);

startActivity(iintent);

}

});

****

**Day 5 :Submit Into next page**

Intent iintent =new Intent(secondActivity.this,MainActivity2.class);

iintent.putExtra("key",name);

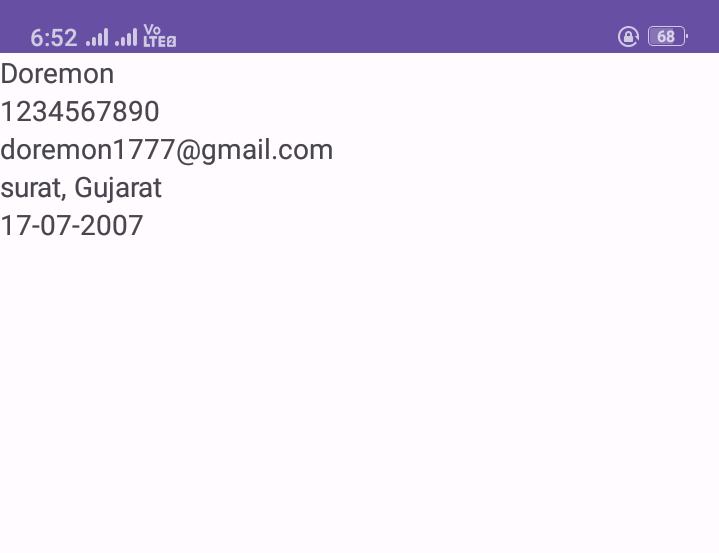
iintent.putExtra("key1",mnumber);

iintent.putExtra("key2",emailadd);

iintent.putExtra("key3",addres);

iintent.putExtra("key4",date);

startActivity(iintent);

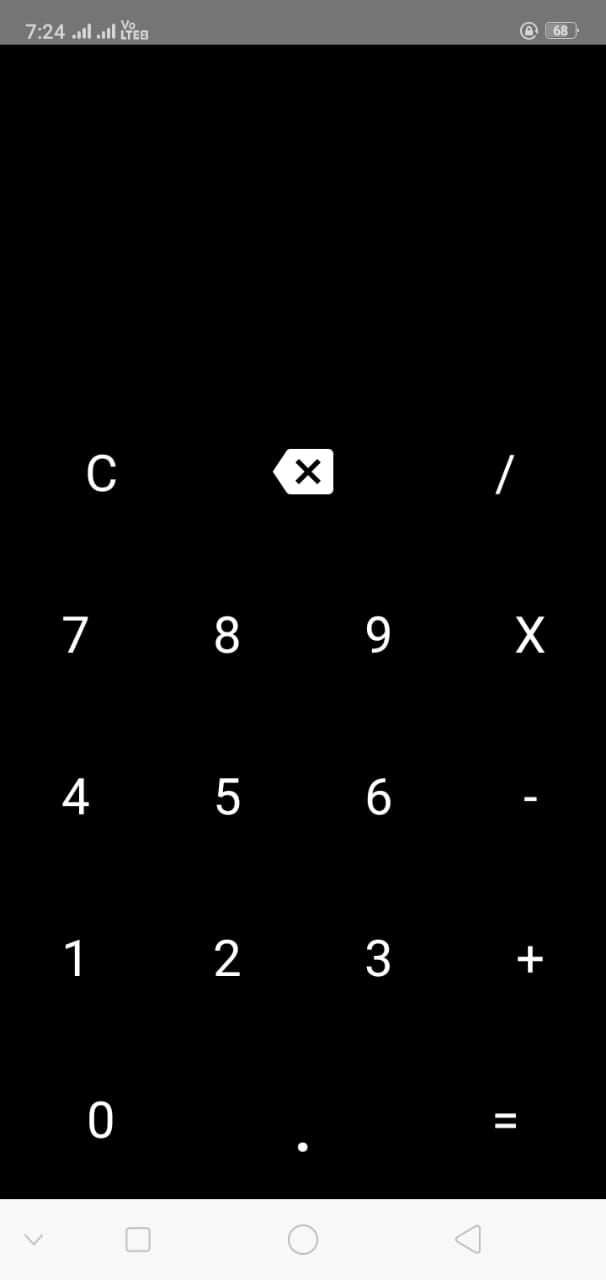
****

**Week 5**

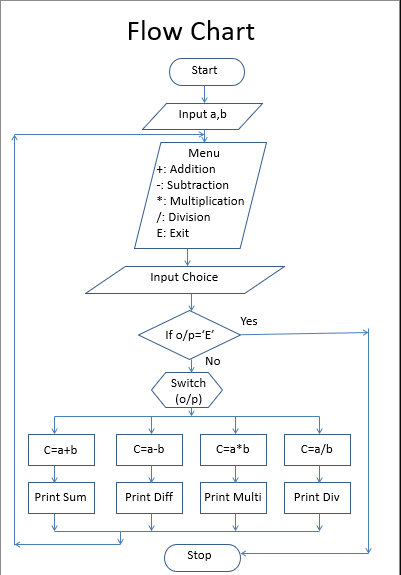
**Day 1 :Calculator**

New

android:layout\_width=”3”

****

**Day 2 : Logic**



**Day 3 : Implementat**ion

* Find all ID in MainActivity
* zero, first, second, third, fourth, fifth, sixth, seventh, eightth, nineth (Button IDs): Used for numeric input from 0 to 9.
* Operation buttons (/, X, -, +) append the operation symbol to tvValue.
* The "Clear" button resets both tvValue and tvResult.
* The app uses the RhinoAndroidHelper class to integrate JavaScript execution within the Android environment.
* Arithmetic expressions are evaluated by converting them into a JavaScript-compatible format (e.g., replacing X with \*).
* he evaluateString method is used to execute the JavaScript code and obtain the result.
* The app also handles exceptions during the JavaScript evaluation to prevent crashes and displays "Error" in tvResult if evaluation fails.
* The app dynamically updates the TextView elements (tvValue and tvResult) based on user input and operations, providing real-time feedback.

**Day 4 :Add Icon**

<application

android:allowBackup="true"

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

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

android:icon="@drawable/tic"

android:label="Tic Tac Toe"

android:roundIcon="@drawable/tic"

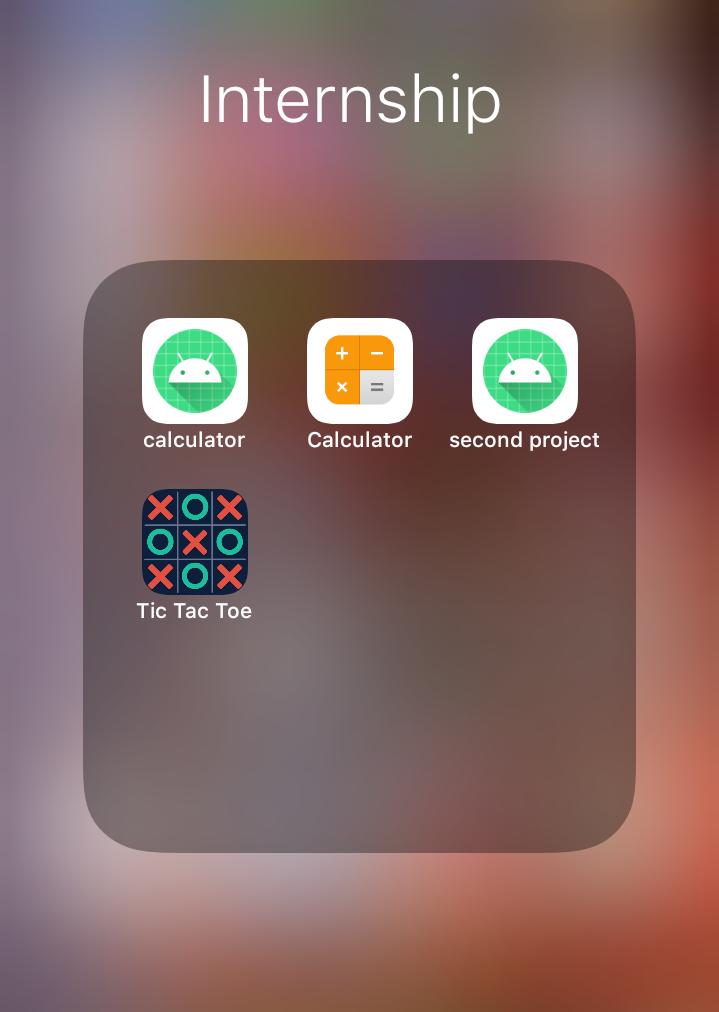
android:supportsRtl="true"

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

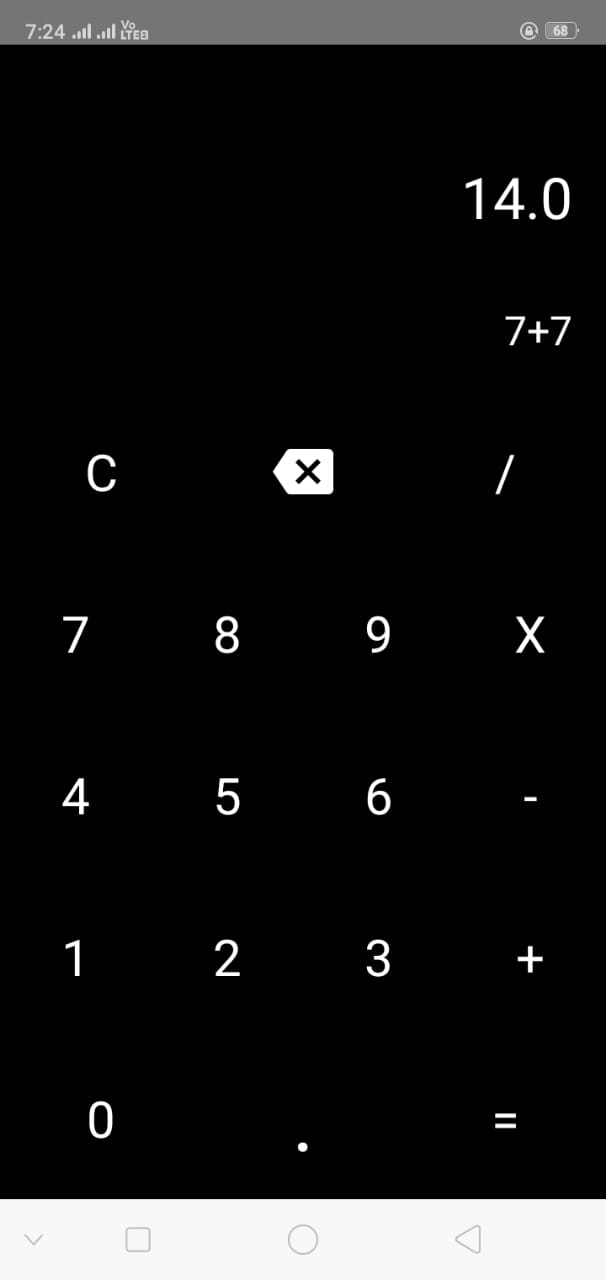
tools:targetApi="31">

</application>





**Day 5 :Run Application**

****

**Week 6**

**Day 1 : To Do Task (MyTask)**

A To-Do Task app is a simple tool that helps people keep track of what they need to get done. Users can create lists of tasks, set reminders, and mark tasks as completed once they’re done.

The app makes it easy to stay organized and remember important tasks, helping users manage their time better and stay on top of their daily responsibilities.

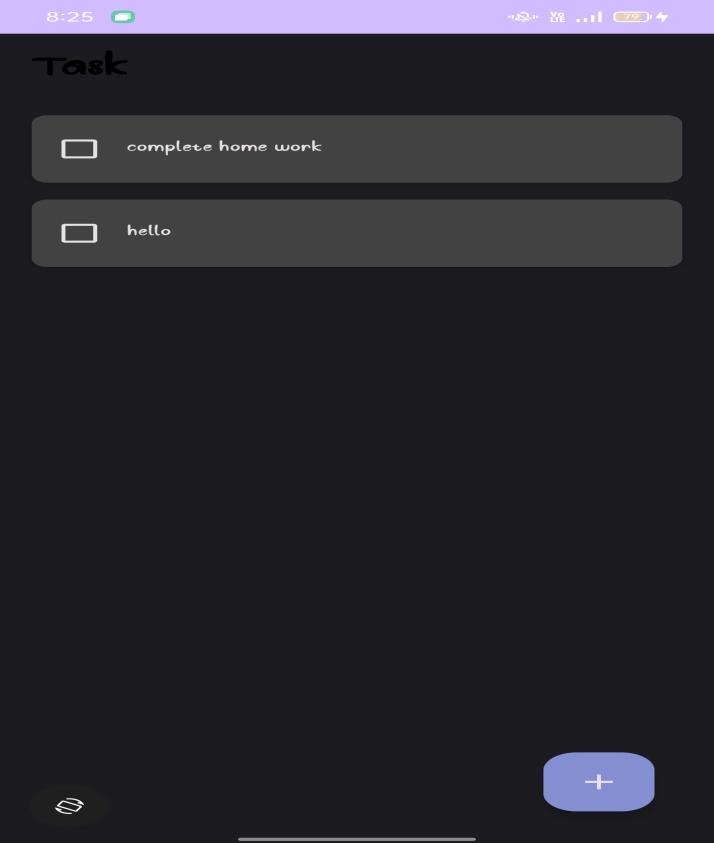
In my app we can Edit Task and also Delete Task very easily

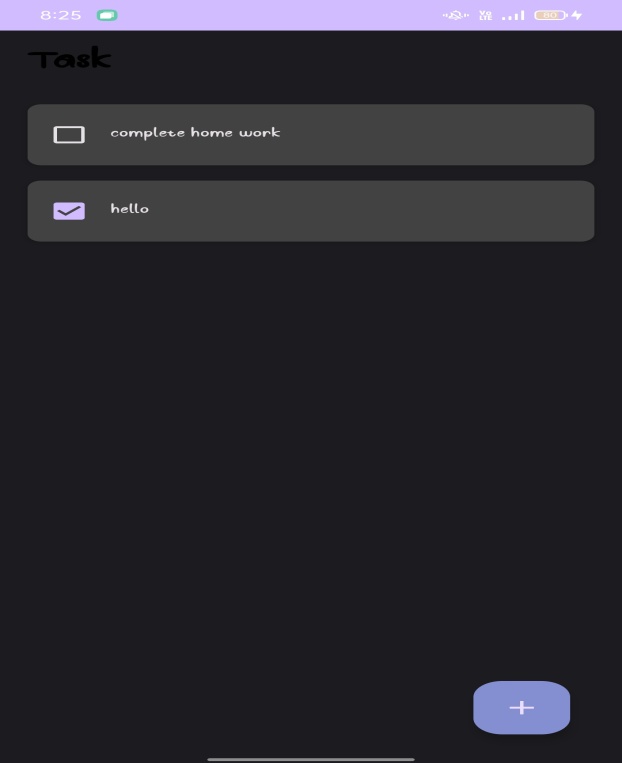
When your task is completed, you can click on the checkbox. By doing this, you'll know that your task is finished.

**Day 2 : Splash Screen**

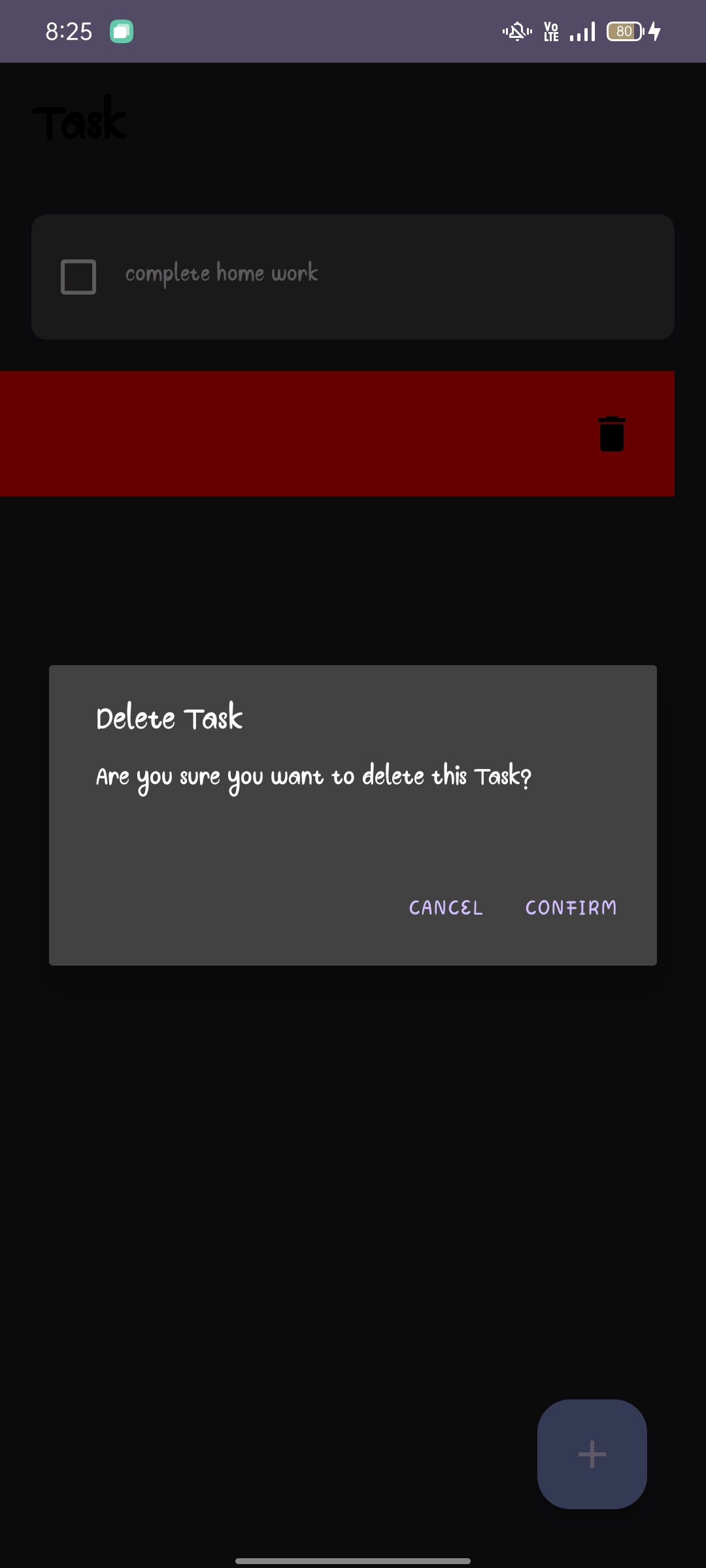
****

**Day 3 : Design**

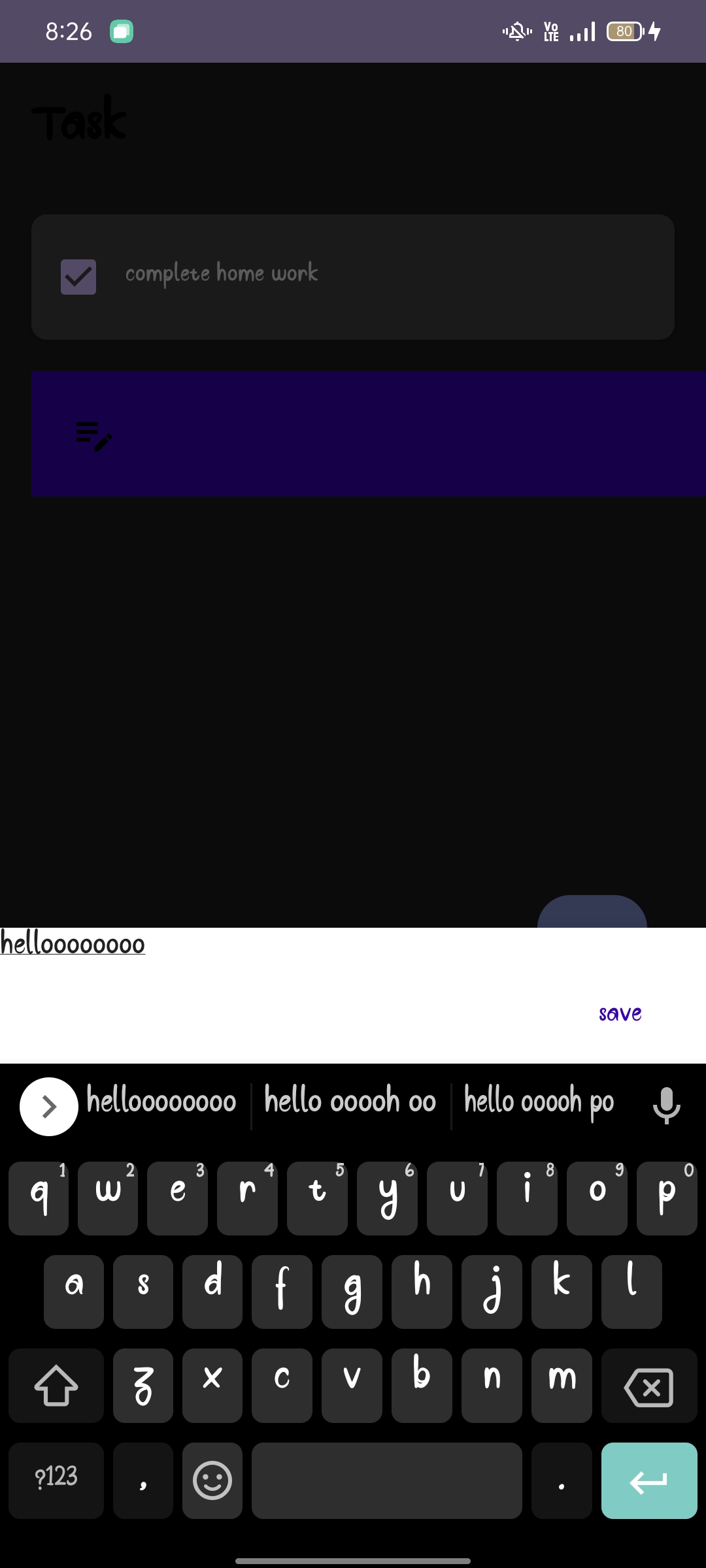
****

****

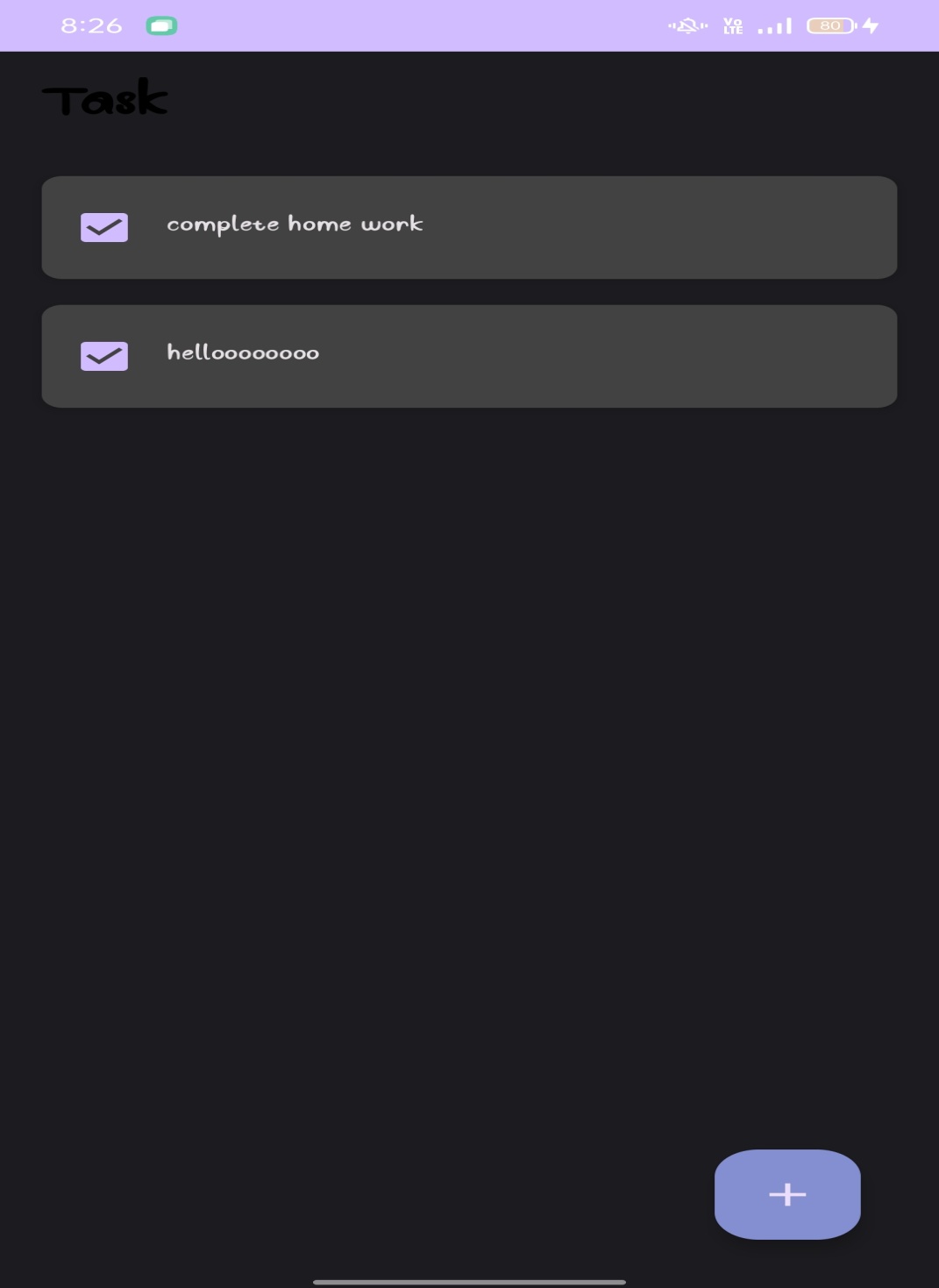
**Day 3 : Delete Task**

****

**Day 4 : Edit Task**

****

**Day 5 : Finally app created**

****