→ To create menu add resource file in the resource folder named menu.

☆ **Summation of two no.s.**

```
var etnm1 : EditText = findViewByID (R.id.txtn1);
var etnm2 : EditText = findViewByID (R.id.txtn2);
var btn : Button = findViewByID (R.id.button);
var tv : TextView = findViewByID (R.id.tv1);
btn.setOnClickListener
{
        var n1 = etnm1.text.toString().toInt();
        "  n2 = etnm2.  "          "          "      ;
        var sum = n1 + n2;
        tv.text = "Addition is :  " + sum.toString();
}
```

☆ **Activity Life Cycle.**

Oncreate :-
- Log.i ("New Activity", "Oncreate");

```
Override fun onStart(){
        super.onStart()
        Log.i ("New Activity", "Onstart called");
}
```

↳ **Similarly,**

onResume(), onPause(), onStop(), onRestart(), onDestr

☆ **Login**

```
var    usm usnm : EditText = find View By ID (R.id. txtnm);
var       psw :     "      =     "            (R.id. txtpswl);
var       bn : Button =      "                (R. id. btn);


bn. setOnClickListener {

        if (usnm. text. toString (). equals ("admin") &&
                pswl. text. toString (). equals ("admin"))
        {
            Toast. makeText (this, "Login Successful",
                Toast. LENGTH_LONG). show ();
        }
}
```

☆ **B** One screen will take student info from user & display that to other screen on submit button click on 1st screen.

→ On one screen (1st) :-

```
var   t1 : EditText = find ViewByID (R.id. txt1);
 "    t2 :     "      =      "          (R.id.txt2)
 "    t3 :     "      =      "          (R. id. txt3)
 "    t4 :     "      =      "          (R.id. txt4)
 "    t5 :     "      =      "          (R. id. txt5)
 "    bn : Button  =      "             (R.id.button)
```

bn. setOn click Listener {

    var id = t1. text. toString():
    " nm = t2. " "
    " brnm = t3. " "
    " mono = t4. " "
    " bd = t5. " "

    var intent = Intent (this, HomePage :: class. java)

                  ↓

                    name of 2nd activity

    intent. putExtra ("ID", id)
    start Activity (intent)

    ↳ repeat it for name, branch name,
        mobile number, birthdate.

}

> For 2nd screen (HomePage)

var tv1 : TextView = findViewByID (R. id. textview)
 " tv2 : " = " (" " ")
 " tv3 : " = "
 " tv4 : " = "
 " tv5 : " = "

var s1 = intent. getStringExtra ("ID"). toString()
 tv1. setText ("ID: " + s1)
↳ Same for s2, s3, s4, s5.

✱ Create 2 screens. Login & Home. If user enters u.n = admin & pass = admin then only home screen should appear. where welcome (u.n) should appear.

→ Login screen :-

```
var  usnm : EditText = findViewByID (R.id. txtom)
 "   pw  :   "    =    "       ( "  ", pw)
 "   bn  : Button  =    "       ( "  ", bn)


bn. seton clicklistener {
    if (usnm. text. tostring() . equals (" admin") &&
        pw. text. tostring(). equals (" admin"))
{  {
        Toast. make Text (this, "Login Successful",
            Toast. Length_long). show()
        var intent = Intent (this, Home Page ::
                                        class. java)
        intent. putstring extra (" Username",
                        usnm. text. tostring())
        Start Activity (intent)
    }

}
```
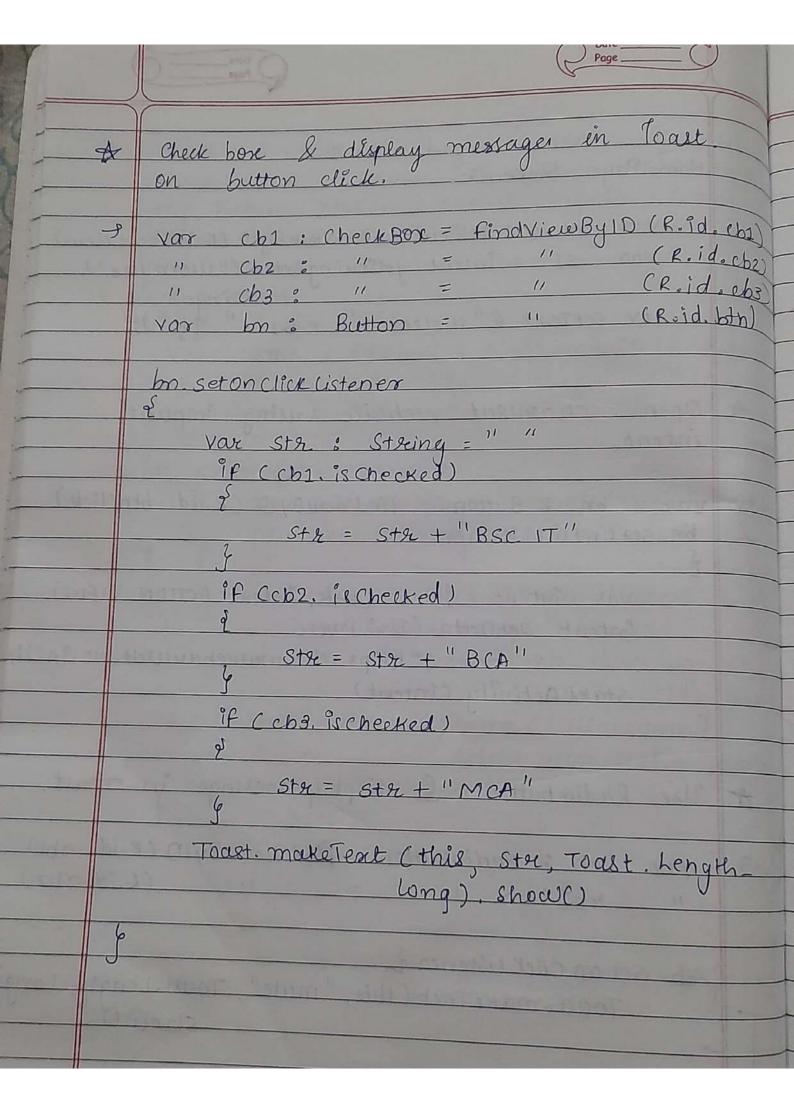
Home Page Screen :-

```
var  tv : TextView = findViewById (R.id. textview)
var  s1  = intent.getStringExtra ("username").
                                    toString()
tv.setText £" welcome " + s1 + " !! ");
```

☆ Open   charusat website   using implicit
intent.

→
```
var   bn : Button = findViewById (R.id. btnclick)
bn.setOnClickListener
{
    var intent = Intent (this, Intent.ACTION_VIEW)
    intent.getData (Uri.Parse
                    ("https://www.charusct.ac.in"));
    startActivity (intent)
}
```

☆ Use Radio button   & display message in Toast.

?
```
var  rcb : RadioButton = findViewById (R.id. rb1)
  "  rcb1 :     "       =     "       (R.id. rb2)
```

```
rcb.setOnClickListener {
    Toast.makeText (this, "male", Toast.length.long
                                      show() }
```

⤷ Same for rcb1 → female.

☆ Check box & display messages in Toast. on button click.

→
```
var    cb1 : CheckBox = findViewByID (R.id.cb1)
 "     cb2 :    "      =     "         (R.id.cb2)
 "     cb3 :    "      =     "         (R.id.cb3)
var    bn  : Button   =     "         (R.id.btn)


bn.setOnClickListener
{
        var str : String = " "
        if (cb1.isChecked)
        {
                str = str + "BSC IT"
        }
        if (cb2.isChecked)
        {
                str = str + "BCA"
        }
        if (cb3.isChecked)
        {
                str = str + "MCA"
        }
        Toast.makeText (this, str, Toast.Length
                        long). show()

}
```

☆ Toggle button & switch print status in Toast (on/off).

→ var tb : ToggleButton = findViewById (R.id.tb)
var sw : Switch = " " (R.id.sw)

sw.setOnCheckedChange.Listener{ _ , checked →
    if (sw.isChecked) {
        Toast.makeText (this, "ON" Toast, Length
                        long).show()
    }
    else {
        Toast (off)
    }
}

↳ Same for toggle button.

⭐ Shared Preferences.

→

```
var etid : EditText = findViewByID (R.id.txtid)
var etps :    "        =        "     (R.id.txtpass)
var bnsave : Button   =        "     (R.id.btnSave)
var bncancel : B "     =        "     (R.id.btncancel)


try {
    var etun1 : E.T  =    "    (R.id.txtid)
    var etpw1 :  "   =    "    (R.id.txtpass)
    val   sharedPreferences =
    getSharedPreferences ("FirstSharedPreference",
                            MODE   PRIVATE)
    var  stringval : String? = sharedPreferences.
                        getString ("name", null)
    var  stringval1 : String? = sharedPreferences.
                        getString ("pw", null)
    etun1.setText (stringval)
    etpw1.setText (stringval1)
}


catch (e: Exception)
{
    Toast.makeText (this, e.toString(), Toast.Length
                        short).show()
}
```

```kotlin
bnsave.setonclick listener
{
        var  etunm : E.T. l = findViewById (R.id.txtid)
        var  etpw :    "    =    "       (R.id.txtpass)
        var  sharedPreferences =
        getShared Preferences ("FirstshardPreferences",
                    MODE_PRIVATE)
        val  editor = sharedPreferences.edit()
        editor.apply
        {
                putString ("name", etunm.text.tostring())
                putString ("pw", etpw.text.tostring())
        }. apply()

        Toast.makeText (this, "saved the Preference",
                Toast.Length_short).show()
y

bnclr.setonclicklistener {
        val preferences =
            getsharedPreferences ("First Shared Preferences",
                    0)

        preferences.edit().remove ("name").commit()
        preferences.edit().remove ("pw").commit()
        ed etid.text.clear()
        etps.text.clear()

}
```

* **Alert Dialog Box.**

→ var btn : Button = findViewByID (R.id. Button)

```
btn. setOnClickListener
{
        val     dialogBuilder = AlertDialog. Builder ( this)
        dialogBuilder. setTitle (" Alert Dialog Example")
        dialogBuilder. setMessage (" Do you want to
                        close it ? ")
        dialog Builder. setPositive Button ("OK",
                Dialog Interface. onClickListener
                        { dialog, id → finish() } )

        dialog Builder. setNegative Button (" NO ",
                Dialog Interface. onClickListener {
                        dialog, id → dialog. cancel() } )

        dialog Builder. show()

}
```

☆ Database connection.
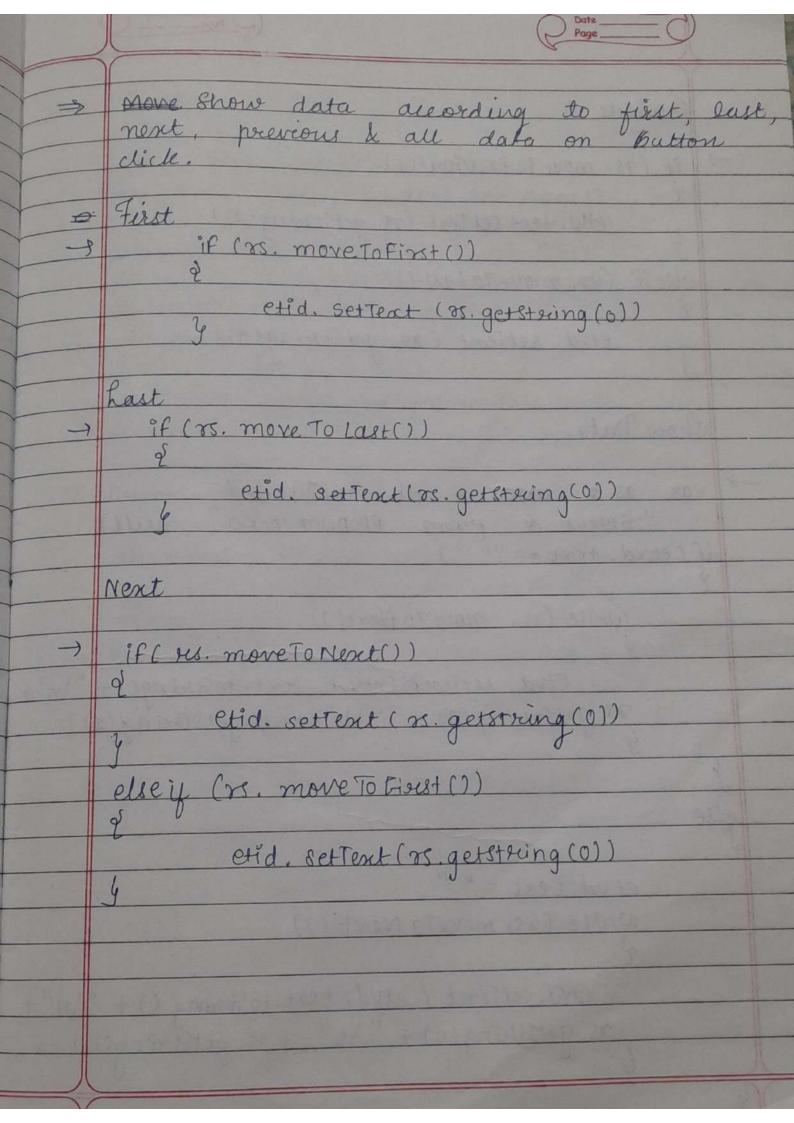
→ Create a new dBHelper class.

```
Class DBHelper (context: Context):
SQLite.OpenHelper (context, "product db", null, 1)
                              ↓
                          database name.
{
        override fun onCreate (p0: sQLiteDatabase)
        {
                # create table query. (2 ways)
                (1.)
                val c.table = " CREATE TABLE PRODUCTINFO
                (PCODE INTEGER PRIMARY KEY, PNAME
                TEXT); "
                p0? execSQL (c table)


                (2)
                p0? execSQL (" CREATE TABLE PRODUCTINFO
                (PCODE INTEGER PRIMARY KEY, PNAME
                TEXT); "
        }


        override fun onUpgrade (p0: sQLiteDatabase?,
                        p1: Int, p2: Int)
        {
                # Drop table if exist query.
                p0?. execSQL (" DROP TABLE IF EXISTS
                        PRODUCTINFO")
                onCreate (p0)
        }
}
```

⇒ Connecting Database class to kotlin file.

```
var helper = DBHelper (application context)
var db : SQLiteDatabase = helper.readable
                                        Database
var rs = db.rawQuery ("select * from
                            PRODUCT INFO", null)
```

var etid & E
On button click listener if you want to insert into database.

```
var cv = ContentValues().
┌ cv.put ("PID", etid.text.to String())
└→ write cv.put as many times as
   the no. of columns in your table.

db.insert ("PRODUCT INFO", null, cv)
                ↓
            table-name
rs.requery()
```

Toast.makeText(this, "PRODUCT ADDED", Toast.
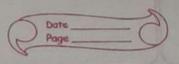   LENGTH LONG).show()

⇒ ~~Move~~ Show data according to first, last, next, previous & all data on button click.

▭ First

→
```
        if (rs. moveToFirst())
        {
                etid. setText (rs. getString (0))
        }
```

Last

→
```
    if (rs. moveToLast())
    {
            etid. setText (rs. getString(0))
    }
```

Next

→
```
    if( rs. moveToNext())
    {
            etid. setText (rs. getString (0))
    }
    elseif (rs. moveToFirst())
    {
            etid. setText (rs. getString (0))
    }
```

Previous

→ if (rs. moveTo Previous())
{
    etid. text setText (rs. getString (0))
}
else if (rs. moveTo Last ())
{
    etid. setText (rs. getString (0))
}

Show Data.

→ var rs : Cursor = db. rawQuery (
    " Select * from PRODUCT INFO ", null)
if (etvd. text == " ")
{
    while (rs. moveTo Next())
    {
        etvd. setText (etvd. text. toString() + "\n" +
        rs. getString (0) + "\t" + rs. getString (1))
    }
}
else
{
    etvd. text = " "
    while (rs. moveTo Next ()
    {
        etvd. setText (etvd. text. toString () + "\n" +
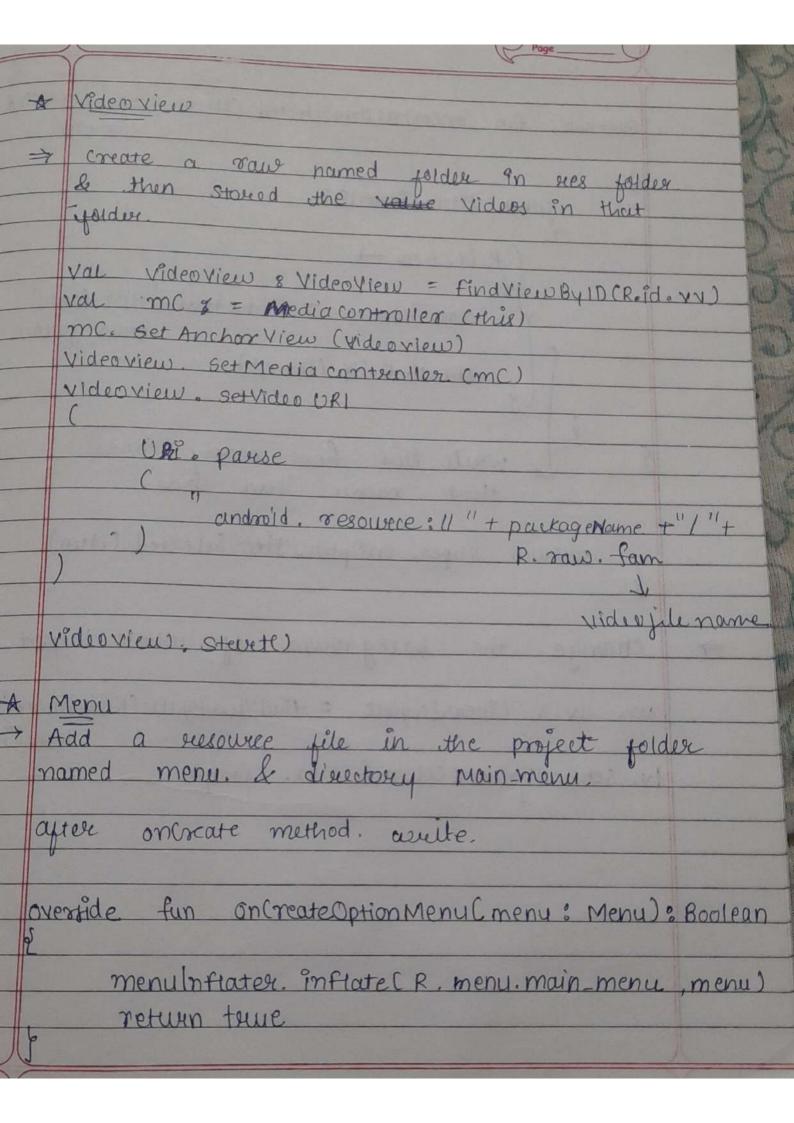   rs. getString (0) + "\t" + rs. getString(1))
    }
}

→ Update the data.

```
var  cv = ContentValues ()
cv. put ("PID", etid.text.toString())

db. update ("PRODUCTINFO", cv, "PID = ?",
            arrayOf (etid.text.toString()))
rs. requery()
if (rs. moveToFirst())
{
        etid. setText (rs. getString(o))
}
```

→ Delete the data.

```
db. delete ("PRODUCT INFO", "PID = ?",
            arrayof (rs. getString (0))]
rs. requery()
Toast. makeText (this, "Record Deleted", Toast. Length-L
                show()
if (rs. moveToFirst ())
{
        etid. setText (rs. getString (0))
}
```

✫ Spinner

```
val    colors : Array <string> = arrayOf
       ("Select  any  color", "Red", "Green", "Blue",
        "Yellow";)
```

⤷ declare this variable globally.

```
var    sp : Spinner = findViewByID (R.id.Spinner);
       to var   sp_value : String
var    adapter : ArrayAdapter <string> =
       ArrayAdapter <String> (this, android.R.layout.
                  select_dialog_item , colors)


Sp. setAdapter (adapter)
sp. onItemSelectedListener = object : AdapterView.
    OnItem Selected Listener
{
       override fun onItemSelected ( parent : AdapterView<*>,
       view : View?, position : Int, id : Long)
       {
          sp_value = sp. SelectedItem. toString()
       }

       override fun onNothingSelected (parent :
          AdapterView <*> ?)
       {
          // TODO ("Not Yet Implemented")
       }
}
```

⟹ if we want to pass spinner value in database
   pass the value stored in sp_value.

☆ **Video view**

⇒ Create a raw named folder in res folder & then stored the value videos in that folder.

```
val videoview : VideoView = findViewByID (R.id.vv)
val mC : = Mediacontroller (this)
mC. set Anchor View (videoview)
videoview. set Media controller (mC)
videoview. set Video URI
(
        URI. parse
            (
               "android. resource: // " + packageName + "/" +
                                        R. raw. fam
                                              ↓
                                        video file name
videoview. start()
```

☆ **Menu**

→ Add a resource file in the project folder named menu. & directory Main-menu.

after oncreate method. write.

```
overide fun onCreateOptionMenu ( menu : Menu) : Boolean
{
        menuInflater. inflate ( R. menu. main_menu , menu)
        return true
}
```

override fun onOptionsItemSelected (item : MenuItem):
Boolean

{

  when (item.itemID)

  {

    R.id.bca →

    {

      var intent = Intent (this, `Bca_form!!`
                     claus. java)

      StartActivity (intent)

    }

  }
     ↳ write this how many times
       that menu item has.

  return super. onOptionsItemSelected (item)

}

→ change the background of whole layout.

var lv : LinearLayout = findViewByID (R.id.linear
               layout)

lv. setBackgroundColor (color. Gray)