**Fragments**

* onCreatView function xml file attached //create a view
* **Load fragment:** getSupportFragmentManager.beginTransaction().replace(R.id.fragmentcontainer, new fragmentName()).commit();
* onviewCreated //when view created
* Toast.makeText(getContext(), “you clicked me”, Toast.LENGH\_SHORT).show();

**Alert**

* new AlertDialog.Builder(getContext()) //if activity then in parmeter activity.this

setTitle(“set Title”)

setMessage(“message”)

.setPositiveButton("ok", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 Toast.*makeText*(getContext(),"ok clicked",Toast.*LENGTH\_SHORT*).show();  
  
 }  
})

.setNegitiveButton(text:”ok”, listener: null)

.setNeutralButton(text:”ok”, listener: null)

.show();

* positive button on right of alert box
* negative button on left of positive button
* neutral button on left of alert box

**Bottom Navigation**

* bottonNavigation.bottomNavigationView //view in xml
* create menu file for bootom navigation from resource file //res folder
* minimum 2 items and maximum 5 items in navigation
* creating items in manu:

<item

android:id= “@+/nav\_item1”

android:icon=”@drawable/name”

android:title=”title”/>

* show manue in xml bottom navigation:

app:manu=”@manu/manuName”

* **default behaviour**:If two or three items in manu, it will show icon and name. if greater then he will show only selected item icon and name else only icon.
* app:lableVisibilityMode=”selected or unlabeled or la bled or auto”
* set onItemSeletedListner() on bottom navigation view. Get selected item id by id=item.getItemid() and by if condition match it with menu item ids.
* Return false in menu item condition.

**Laucher activity**

Manifest folder

Exported=true

<intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
</intent-filter

**Toolbar remove:**

Values folder then theme folder and set NOActionBar in box theme xml files

**Drawer:**

* drawerLayout will be use in drawer xml file
* openDrawer=”start” for preview of drawer in screen //drawer layout
* two parts coordinatorLayout and naviagtionview
* CorrdinatorLayout contain toolbar and maincontainer where fragments may run.
* Naviagationview(manu item option)
* App:layout\_behavior=”scrolling” //to set framelayout below to appbar
* Create menu just like bottomnavigation bar but set all item in group tags. <group checkableBehavior=”single”>

<item

android:id= “@+/nav\_item1”

android:checked=”true”

android:icon=”@drawable/name”

android:title=”title”/>

</group>

* Set headerLayout on drawer: headerLayout=”@layout/name”
* setSupportActioBar(toolbar)
* **code for drawer toggle icon:**

1-actionBarDrawerToggle toggle=new actionBarDrawerToggle(activity.this, drawer,toolbar, R.string.open\_drawer, R.string.close\_drawer);

2- drawer addDrawerListner(toggle)

3-toogle.syncState();

* setNavigationItemSlectedListner same as bottomNavigationBar

**Tabs:**

* toolbar , tabbar, viewPager three sections in tabbar screen
* TabLayout view for tabBar in xml
* In viewPager ViewPager view will be use
* Create a class and extend with fragmentpagerAdapter in which one constructor and two other functions getcount and getItem.

In get count mention number of tabs and in getItem mention tab fragments. And getPageTitle function for tab title text.

e.g getPageTitle(int position){

if (position==0){ return “latest”}} same in getItem function but return new latestFragment()

* **Now initialize adapter:**

TabsAdpater adapter=new TabsAdapter(getSupportFragmentManager())

viewPager.setAdpater(adapter)

tabLayout.setUpWithViewPager(viewPager)

* tabTextColor, tabSelectedtextColor, tabIndicatorColor,tabIndicatorHeight
* app:elevation=0dp //remove appbar shadow

**Wallpaper**

* constraintDimensionRatio=”H,1:1”
* scaleType=”centerCrop”
* View itemView=LayoutInflator.from(parent.getContext()).inflate(R.layout.item\_layout\_wallpaper, parent, false);
* Picasso.get().load(currentItem.ImgUrl).placeholder(R.drawable.placeholder).into(holder.ivwallpaper);
* Volly.newRequestQueue(getContext()).add(request)
* adapter.notifyDataSetChanged()
* photoView liberary for imageZoom //jetpack repository
* SET\_WALLPAPER permission set in maniest
* 1- WallpaperManager manager=(WallpaperManager) getSystemService(WALLPAPER\_SERVICE)
* 2- manager.setBitmap(bitmap)
* gson liberary use for fetching data from url
* @SerializedName(“webformatUrl”)
* Gson gson=new Gson()
* className w=gson.fromJson(wallPaperObject.toString(),className.class)
* list.add(w)
* arrayList<className> temlist=gson.fromJsom(array.toString(),new TypeToken<ArryList<className>>(){}.getType());
* wallPaperList.addAll(temlist)

**Local DataBase**

* create class ‘DbHelper’ and extends with SQLiteOpenHelper
* private static final String DB\_NAME=’notes\_db’
* private static final int DB\_VERSION=1
* private static final String TABLE\_NOTES=’tbl\_notes’;
* private static final CREATE\_NOTES-TABLE= “CREATE TABLE”+ TABLE\_NOTES+”(id INTEGER AUTOINCREAMENT PRIMARY KEY,title TEXT, create\_time INTERGER)
* In onCreate Function execute the created table : db.execSQL(CREATE\_NOTES\_TABLE);
* In onUpgrade fun: db.execSQL(“drop table if exist”+ TABLES\_NOTES); 2- onCreate(db)
* Create model class for not\_app Note
* 4 functions in db\_helper class add, update, delete,getAllNotes
* Public long addNote(Note not){ }, public int deleteNote(int id), public ArrayList<Note> getAllNotes{}
* **Public long addNotet(Note note){ ContentValues cv=new ContentValues(); cv.put(“id”,note.getId); SQLiteDatabase db= getWriteableDatabase(); return db.insert(TABLE\_NOTES, null, cv);}**
* Public lng UpdateNote(Note note){ return db.update(TABLE\_NOTES, cv, “id=”+ note.getId(),null)}
* Public int deleteNote(int id){ return db.delete(TABLE\_NOTES, “id=”+id ,null)}
* Public ArrayList<Note> getAllNotes(){

ArrayList<Note> noteslist=new ArrayList<>();

**SQLiteDatabase db= getReradableDatabase();**

**Cursor cursor= db.rawQuery(“Select \* from”+NOTE\_TABLE, null);**

**Int idIndex= cursor.getColumnIndex(“id”);**

**While(cursor.moveToNext()){**

**Long id=cursor.getlong(idIndex)**

**…**

**Note n=new Note();**

**n.setId(id);**

**…**

**notesList.add(n)**

**}**

**cusor.close();**

**retrun notesLIst();**

* System.currentTimeMillis()
* SimpleDateFormat sdf =new SimpleDateFormat(“MMM dd, hh:mm a”,Local.US)
* String datetime=sdf.format(currentItem.getCreatedTime())
* Protected void onResume(){ super.onResume(); fetchnotfor mDb();}
* To putextra object need to write implements Serializable onword to Model class name
* In receive Intent selectintent=(Note) receivedIntent.getSerializabeExtra(“note”)
* lineSpaceingExtra=”2dp”

**Server:**

* phpMyAdmin ubuntu nginix digital occeason
* ubuntu 16 for free
* letsencrypt nginix digital occeason //ssl free

**DB configuration php file:**

$db\_host=”localhost”; $db\_user=”root”; …

$con= mysqli\_connect($db\_host,$db\_user,$db\_password,$db\_name)

If($con){echo “success”}else{echo “error:” .mysqli\_connect\_error()}

**Categories.php**

Include(‘config.php’)

$query=”SELECT \* FROM category\_table WHERE status=1”

$result=mysqli\_query($con,$query)

$categories[]

While($row=mysqli\_fetch\_assoc($result)){ $categories[]=$row}

Echo json\_encode($catetories)

**Products.php**

$catId=$\_GET[‘cat\_id’]

$query=”SELECT \* FROM category\_table WHERE status=1 AND cat\_id=$catId”

**UserAccount**

$result=mysqli\_query($con, “SELECT username FROM table WHERE username=$username)

mysqli\_num\_rows($result)

$id=mysqli\_insert\_id($con) //last inserted row id

**BottomSheetDialog:**

BottomSheetDialog sheetdialog=new BottomSheetDialog():

Sheetdialog.setContentView(R.Layout.layoutName);

Sheetdialog.show();

Button btnAdd=sheetdialog.findViewById(R.id.btnAdd)

Same for btnRemove,btnaddToCart,tvquantity

Actionitembadge for cartitem on toobar

getCartItemCount(){SQliteDatabase db=new getReadableDatabase();

Cursor cursor=db.rawquery(“SELECT id FROM” +car\_table, null)

Int count=cursor.getCount();

getManuInflator().inflate(R..manu.main\_manu, manu) //for actionItemBridge

(String.format(local.US, format: “Rs . %s”,total)

R.string.volley\_error