



L OVELY
P ROFESSIONAL
U NIVERSITY

Continuous Assessment

Submitted By

Darshan

11912717

Section: KO105

Roll No.: B52

BTech CSE Hons.

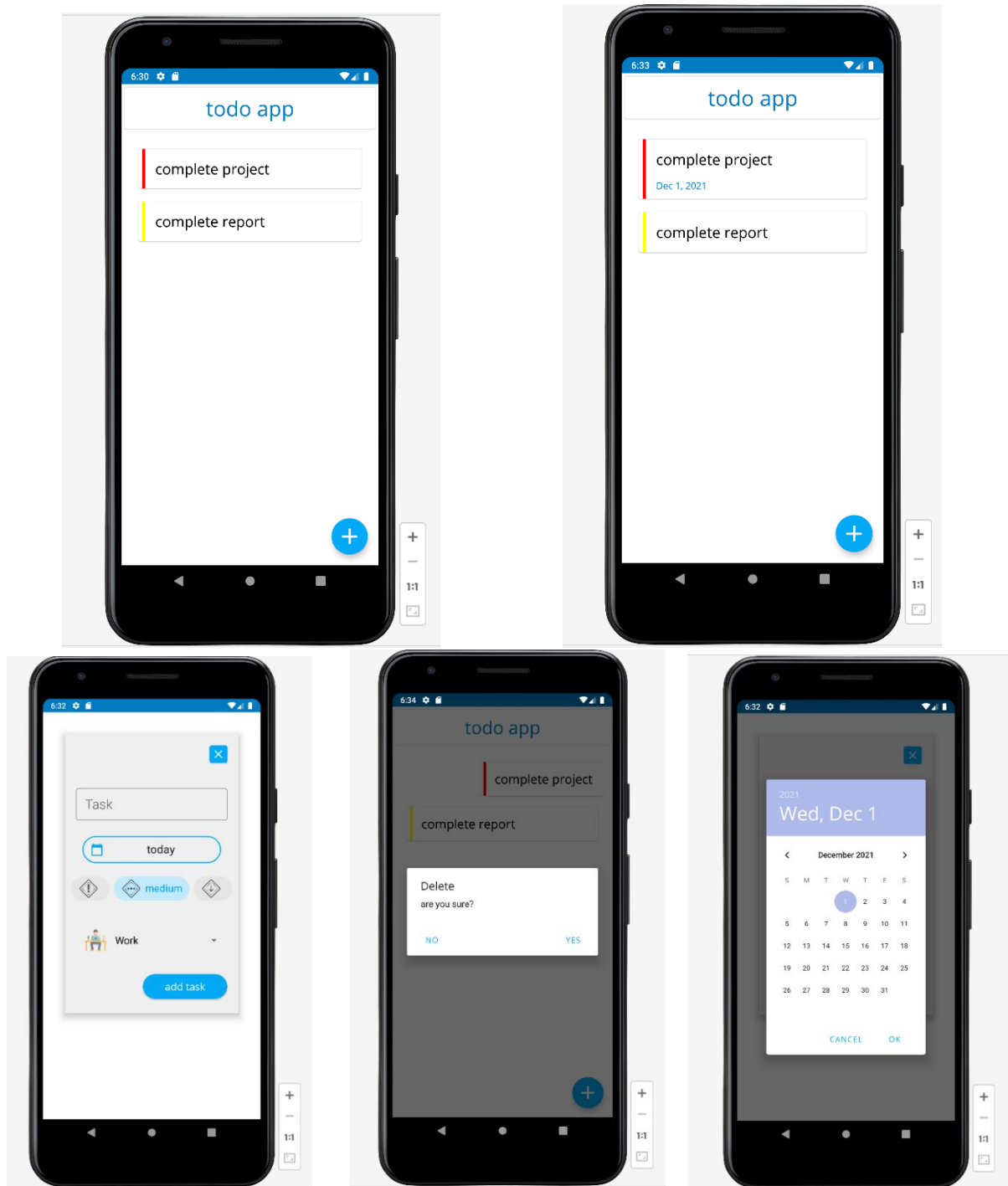
To

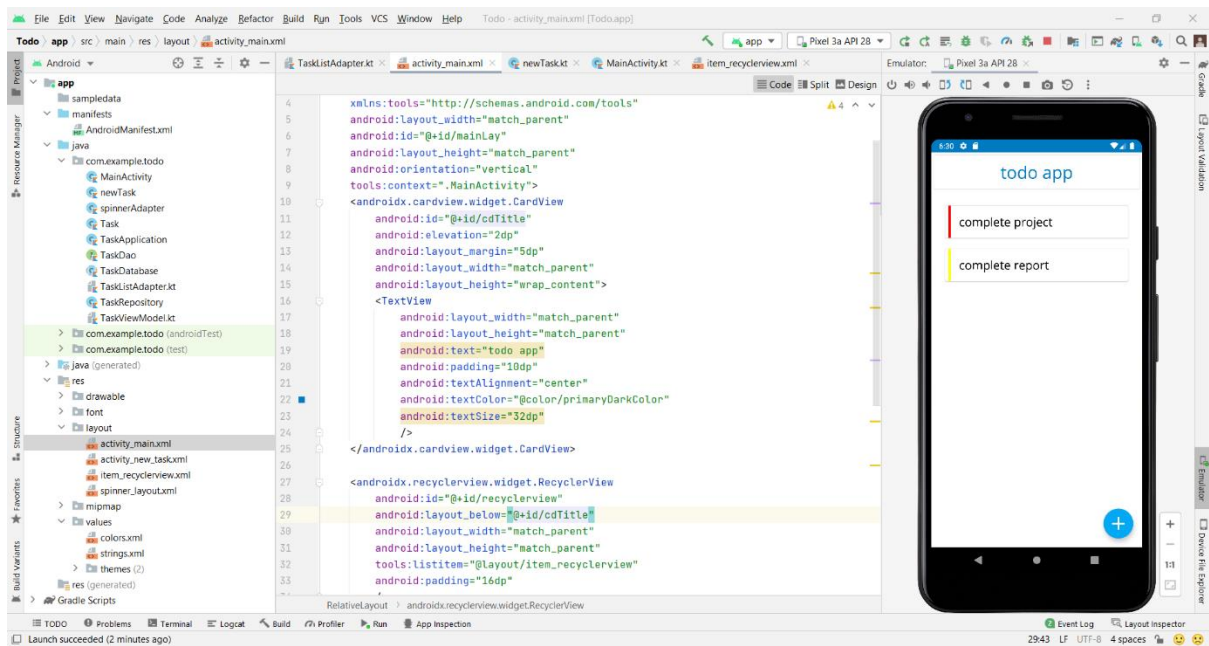
Shailja Sharma Ma'am

School of Computer Science & Engineering

Lovely Professional University, Phagwara

Todo App





This is a simple todo app. In this app user can add tasks and after completing he can delete the task.

I used room database to store the tasks locally.

The tasks will order according to the priority

It has 3 priorities

- 1)high (with red color band)
- 2)medium (with yellow)
- 3)low (green)

While adding task user has to mention date (deadline)

All the items will be displayed in homescreen.

When you tap the item it will show date.

When you swipe it will ask to delete.

Activity_main

```
package com.example.todo

import android.app.Activity
import android.app.AlertDialog
import android.content.Intent
import android.graphics.*
import android.os.Bundle
import android.view.View
import android.widget.CheckBox
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AppCompatActivity
import androidx.lifecycle.observe
import androidx.recyclerview.widget.ItemTouchHelper
import androidx.recyclerview.widget.ItemTouchHelper.LEFT
import androidx.recyclerview.widget.ItemTouchHelper.RIGHT
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.example.todo.newTask.Companion.EXTRA_PRIORITY
import com.example.todo.newTask.Companion.EXTRA_TITLE
import com.google.android.material.floatingactionbutton.FloatingActionButton
import com.google.android.material.snackbar.Snackbar
import java.lang.Exception
import android.graphics.Paint
import android.widget.TextView
import com.example.todo.newTask.Companion.EXTRA_DATE

class MainActivity : AppCompatActivity() {
    lateinit var adapterTask : TaskListAdapter
    lateinit var recyclerView: RecyclerView
    private val newTaskActivityResultCode = 1

    private val taskViewModel: TaskViewModel by viewModels {
        TaskViewModelFactory((application as TaskApplication).repository)
```

```

    }
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        recyclerView = findViewById<RecyclerView>(R.id.recyclerview)
        adapterTask = TaskListAdapter()
        recyclerView.adapter = adapterTask
        recyclerView.layoutManager = LinearLayoutManager(this)

        setUpListener()

        taskViewModel.allTasks.observe(this) { tasks ->
            tasks?.let { adapterTask.submitList(it) }
        }

        val addTask = findViewById<FloatingActionButton>(R.id.addTask)
        addTask.setOnClickListener() {
            val intent = Intent(this@MainActivity, newTask::class.java)
            startActivityForResult(intent, newTaskActivityRequestCode)
        }
    }

    private fun setUpListener() {
        ItemTouchHelper(object : ItemTouchHelper.SimpleCallback(0, LEFT or
RIGHT) {
            override fun onMove(
                recyclerView: RecyclerView,
                viewHolder: RecyclerView.ViewHolder,
                target: RecyclerView.ViewHolder
            ): Boolean {
                return false
            }

            override fun onSwiped(viewHolder: RecyclerView.ViewHolder,
direction: Int) {
                val builder = AlertDialog.Builder(this@MainActivity)
                builder.setTitle("Delete")
                builder.setMessage("are you sure?")
                builder.setNegativeButton("yes") {
                    dialog, which->
                    val task =
adapterTask.getTaskAt(viewHolder.adapterPosition)
                    taskViewModel.delete(task)
                }
                builder.setNeutralButton("no") {
                    dialog, which->
                    adapterTask.notifyItemChanged(viewHolder.getAdapterPosition())
                }
            }
        })
    }

```

```

        builder.setCancelable(false)
        val ad:AlertDialog = builder.create()
        ad.show()
    }

    override fun onChildDraw(
        c: Canvas,
        recyclerView: RecyclerView,
        viewHolder: RecyclerView.ViewHolder,
        dx: Float,
        dy: Float,
        actionState: Int,
        isCurrentlyActive: Boolean
    ) {
        try {
            val icon: Bitmap
            if (actionState == ItemTouchHelper.ACTION_STATE_SWIPE)
            {
                val itemView = viewHolder.itemView
                val height = itemView.bottom.toFloat() -
itemView.top.toFloat()
                val width = height / 10
                viewHolder.itemView.translationX = dx / 3

            } else {
                super.onChildDraw(
                    c,
                    recyclerView!!, viewHolder, dx, dy,
                    actionState, isCurrentlyActive
                )
            }
        } catch (e: Exception) {
            e.printStackTrace()
        }
    }

    }).attachToRecyclerView(recyclerView)
}

    override fun onActivityResult(requestCode: Int, resultCode: Int, data:
Intent?) {
        super.onActivityResult(requestCode, resultCode, data)
        if(data != null && requestCode == newTaskActivityRequestCode &&
resultCode == Activity.RESULT_OK) {
            val title: String? = data.getStringExtra(EXTRA_TITLE)
            val priority: Int = data.getIntExtra(EXTRA_PRIORITY, -1)
            val dateS:String? = data.getStringExtra(EXTRA_DATE)
            title?.let { dateS?.let { it1 -> Task(it,priority, it1) }
}?.let { taskViewModel.insert(it) }
            Toast.makeText(this, "Note inserted!",
Toast.LENGTH_SHORT).show()

        }else {
            Toast.makeText(this, "Note not saved!",
Toast.LENGTH_SHORT).show()
        }
    }
}

```

newTask.kt

```
package com.example.todo

import android.app.Activity
import android.app.DatePickerDialog
import android.content.Context
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.text.TextUtils
import android.view.View
import android.view.inputmethod.InputMethodManager
import android.widget.AdapterView
import android.widget.DatePicker
import android.widget.Spinner
import android.widget.TextView
import com.google.android.material.button.MaterialButton
import com.google.android.material.chip.Chip
import com.google.android.material.chip.ChipGroup
import com.google.android.material.circularreveal.CircularRevealRelativeLayout
import com.google.android.material.floatingactionbutton.FloatingActionButton
import com.google.android.material.textfield.TextInputEditText
import java.util.*

class newTask : AppCompatActivity() {
    internal var category = arrayOf("Work", "Personal", "Hustle", "Other")
    internal var images =
        intArrayOf(R.drawable.work, R.drawable.personal, R.drawable.hustle,
R.drawable.other)

    lateinit var datePick: TextView
    var day = 0
    var month: Int = 0
    var year: Int = 0
    var dateString: String? = ""

    lateinit var editWordView : TextInputEditText

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_new_task)

        var expand =
findViewById<CircularRevealRelativeLayout>(R.id.expand)
        var addTask = findViewById<FloatingActionButton>(R.id.addTask)
        var sub = findViewById<MaterialButton>(R.id.submitBtn)
        editWordView = findViewById<TextInputEditText>(R.id.edit_word)
        var closeBtn = findViewById<MaterialButton>(R.id.closeBtn)
        datePick = findViewById<Chip>(R.id.datePick)
        var high = findViewById<Chip>(R.id.chip1)
        var medium = findViewById<Chip>(R.id.chip2)
        var low = findViewById<Chip>(R.id.chip3)
        var chipGroup = findViewById<ChipGroup>(R.id.priority)
```

```

var monthName = arrayListOf<String>(
    "Jan", "Feb",
    "Mar", "Apr", "May", "Jun", "Jul",
    "Aug", "Sept", "Oct", "Nov",
    "Dec"
)

val calendar: Calendar = Calendar.getInstance()
day = calendar.get(Calendar.DAY_OF_MONTH)
month = calendar.get(Calendar.MONTH)
year = calendar.get(Calendar.YEAR)
dateString = String.format("%s %d, %d", monthName[month], day, year)

datePick.setText("today")

closeBtn.setOnClickListener() {
    val imm = getSystemService(Context.INPUT_METHOD_SERVICE) as?
InputMethodManager
    imm?.hideSoftInputFromWindow(it.windowToken, 0)
    finish()
}

datePick.setOnClickListener() {
    val calendar: Calendar = Calendar.getInstance()
    day = calendar.get(Calendar.DAY_OF_MONTH)
    month = calendar.get(Calendar.MONTH)
    year = calendar.get(Calendar.YEAR)

    val datePickerDialog =
        DatePickerDialog(this, object :
DatePickerDialog.OnDateSetListener {
            override fun onDateSet(
                view: DatePicker?,
                year: Int,
                month: Int,
                dayOfMonth: Int
            ) {
                //                dateString = String.format("%s %d,
%d", monthName[month], dayOfMonth, year)
                if (dayOfMonth == day) {
                    datePick.setText("today")
                } else {
                    dateString = String.format("%s %d,
%d", monthName[month], dayOfMonth, year)
                    datePick.setText(
                        String.format(
                            "%s %d, %d",
                            monthName[month],
                            dayOfMonth,
                            year
                        )
                    )
                }
            }
        }, year, month, day)

    datePickerDialog.show()
}

```



```

sub.setOnClickListener() {
    val replyIntent = Intent()
    if (TextUtils.isEmpty(editWordView.text)) {
        setResult(Activity.RESULT_CANCELED, replyIntent)
    } else {
        //        val word = editWordView.text.toString()
        //        replyIntent.putExtra(EXTRA_REPLY, word)
        //        setResult(Activity.RESULT_OK, replyIntent)
        var taskT = editWordView.text.toString()
        //        when(chipGroup.checkedChipId) {
        //
        //            }
        var ds = dateString.toString()
        var prioId = chipGroup.checkedChipId
        replyIntent.putExtra(EXTRA_TITLE, taskT)
        replyIntent.putExtra(EXTRA_PRIORITY, prioId)
        replyIntent.putExtra(EXTRA_DATE, ds)
        setResult(Activity.RESULT_OK, replyIntent)

    }
    finish()

}

high.setOnClickListener() {
    if (high.isChecked) {
        high.setText("high")
        medium.setText("")
        low.setText("")
    } else {
        high.setText("")
    }
}

medium.setOnClickListener() {
    if (medium.isChecked) {
        high.setText("")
        medium.setText("medium")
        low.setText("")
    } else {
        medium.setText("")
    }
}

low.setOnClickListener() {
    if (low.isChecked) {
        high.setText("")
        medium.setText("")
        low.setText("low")
    } else {
        low.setText("")
    }
}

}

val spin = findViewById<View>(R.id.categorySpinner) as Spinner
spin.onItemSelectedListener = object :
AdapterView.OnItemSelectedListener {
    override fun onItemSelected(
        parent: AdapterView<*>,
        view: View,
        position: Int,

```

```

        id: Long
    ) {

    }

    override fun onNothingSelected(parent: AdapterView<*>) {

    }
}
val spinAdapter = spinnerAdapter(applicationContext, images,
category)
spin.adapter = spinAdapter
}

companion object {
    const val EXTRA_REPLY = "com.example.android.wordlistsql.REPLY"
    const val EXTRA_ID = " com.example.todo.EXTRA_ID"
    const val EXTRA_TITLE = " com.example.todo.EXTRA_TITLE"
    const val EXTRA_DESCRIPTION = "
com.example.todo.EXTRA_DESCRIPTION"
    const val EXTRA_PRIORITY = " com.example.todo.EXTRA_PRIORITY"
    const val EXTRA_DATE = "com.example.todo.EXTRA_DATE"
}
}

```

Spin Adapter.kt

```

package com.example.todo

import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.BaseAdapter
import android.widget.ImageView
import android.widget.TextView

class spinnerAdapter(internal var context: Context, internal var images:
IntArray, internal var
categoryName: Array<String>): BaseAdapter() {

    internal var inflater: LayoutInflater
    init {
        inflater = LayoutInflater.from(context)
    }
    override fun getCount(): Int {
        return images.size
    }

    override fun getItem(position: Int): Any? {
        return null
    }

    override fun getItemId(position: Int): Long {
        return 0
    }
}

```

```

        override fun getView(i: Int, view: View?, viewGroup: ViewGroup): View {

            val view = inflater.inflate(R.layout.spinner_layout, null)
            val icon = view.findViewById<View>(R.id.imageView) as ImageView?
            val names = view.findViewById<View>(R.id.textView) as TextView?
            icon!!.setImageResource(images[i])
            names!!.text = categoryName[i]
            return view
        }
    }
}

```

TaskListAdapter.kt

```

package com.example.todo

import android.content.res.Resources
import android.graphics.Color
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.LinearLayout
import android.widget.TextView
import android.widget.Toast
import androidx.core.view.isVisible
import androidx.recyclerview.widget.ListAdapter
import androidx.recyclerview.widget.RecyclerView
import com.example.todo.TaskListAdapter.TaskViewHolder
import androidx.recyclerview.widget.DiffUtil
import android.widget.CheckBox
import androidx.core.content.ContextCompat
import androidx.core.content.res.ResourcesCompat.getColor

class TaskListAdapter : ListAdapter<Task, TaskViewHolder>(diffCallback) {

    lateinit var tex:TextView

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
TaskViewHolder{
        val view = LayoutInflater.from(parent.context)
            .inflate(R.layout.item_recyclerview, parent, false)
        return TaskViewHolder(view)
    }

    override fun onBindViewHolder(holder: TaskViewHolder, position: Int) {
        with(getItem(position)) {
            holder.taskItemView.text = task
            //            holder.taskPrio.text = priority.toString()

            holder.itemView.setOnClickListener() {
                holder.textDate.visibility = if (holder.textDate.visibility
== View.VISIBLE) View

```

```

        .GONE else View.VISIBLE
        holder.textDate.text = dates
    }

    when(priority) {
        R.id.chip1 ->
        holder.taskPrio.setBackgroundResource(R.drawable.red)
        R.id.chip2 ->
        holder.taskPrio.setBackgroundResource(R.drawable.yellow)
        R.id.chip3 ->
        holder.taskPrio.setBackgroundResource(R.drawable.green)
    }

    }

    }

    fun getTaskAt(position: Int) = getItem(position)

    inner class TaskViewHolder(itemView: View) :
        RecyclerView.ViewHolder(itemView) {
        val taskItemView: TextView = itemView.findViewById(R.id.textView)
        val taskPrio : LinearLayout = itemView.findViewById(R.id.prio)
        val textDate : TextView = itemView.findViewById(R.id.hold)

    }

}

//fun onCheckboxClicked(isChecked: Boolean, holder: TaskViewHolder) {
//    if (isChecked) {
//        holder.taskItemView.setTextColor(Color.parseColor("#ffffff"))
//    } else {
//        holder.taskItemView.setTextColor(Color.parseColor("#000000"))
//    }
//}

private val diffCallback = object : DiffUtil.ItemCallback<Task>() {
    override fun areItemsTheSame(oldItem: Task, newItem: Task) = oldItem.id
    == newItem.id

    override fun areContentsTheSame(oldItem: Task, newItem: Task) =
        oldItem.task == newItem.task
        && oldItem.priority == newItem.priority
}

```

Activity_main

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:id="@+id/mainLay"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">
<androidx.cardview.widget.CardView
    android:id="@+id/cdTitle"
    android:elevation="2dp"
    android:layout_margin="5dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="todo app"
        android:padding="10dp"
        android:textAlignment="center"
        android:textColor="@color/primaryDarkColor"
        android:textSize="32dp"
    />
</androidx.cardview.widget.CardView>

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recyclerview"
    android:layout_below="@+id/cdTitle"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:listitem="@layout/item_recyclerview"
    android:padding="16dp"
/>

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/addTask"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    app:fabSize="normal"
    app:maxImageSize="40dp"
    android:backgroundTint="@color/primaryColor"
    android:src="@drawable/ic_baseline_add_24"
    android:layout_margin="16dp"
    app:tint="@color/white"/>

</RelativeLayout>

```

Activity newTask

```
<?xml version="1.0" encoding="utf-8"?>
<com.google.android.material.circularreveal.CircularRevealRelativeLayout
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_margin="40dp"
    android:elevation="10dp"
    android:background="#EEEEEE"
    android:id="@+id/expand"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    tools:context=".newTask">
    <com.google.android.material.button.MaterialButton
        style="@style/Widget.MaterialComponents.Button"
        app:iconTint="@color/white"
        app:cornerRadius="4dp"
        app:elevation="4dp"
        android:insetLeft="0dp"
        android:insetTop="0dp"
        android:insetRight="0dp"
        android:insetBottom="0dp"
        app:iconGravity="textStart"
        app:iconPadding="0dp"
        android:id="@+id/closeBtn"
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:layout_alignParentEnd="true"
        android:layout_margin="20dp"
        app:icon="@drawable/ic_baseline_close_24"/>

    <com.google.android.material.textfield.TextInputLayout

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense"
        android:id="@+id/taskLayout"
        android:layout_below="@+id/closeBtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:hint="Task"
        app:endIconMode="clear_text">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/edit_word"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:fontFamily="sans-serif-light"
            android:inputType="textAutoComplete"
            android:textSize="24dp"
            />

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.chip.Chip
        android:id="@+id/datePick"
        android:layout_width="250dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/taskLayout"
        android:layout_centerHorizontal="true"
```

```

        android:minHeight="65dp"
        android:padding="10dp"
        android:textAlignment="center"
        android:textSize="20dp"
        app:chipBackgroundColor="#EFEFEF"
        app:chipIcon="@drawable/ic_baseline_calendar_today_24"
        app:chipIconTint="@color/primaryColor"
        app:chipStrokeColor="@color/primaryColor"
        app:chipStrokeWidth="2dp"
        app:iconStartPadding="10dp" />

<com.google.android.material.chip.ChipGroup
    android:id="@+id/priority"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/datePick"
    android:layout_centerHorizontal="true"
    android:layout_margin="8dp"
    app:singleSelection="true">

    <com.google.android.material.chip.Chip
        android:id="@+id/chip1"
        style="@style/Widget.MaterialComponents.Chip.Choice"
        android:minHeight="60dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:chipIcon="@drawable/high_priority"
        app:iconStartPadding="10dp"
        app:chipIconSize="35dp"
        android:textSize="18dp"
        app:chipIconVisible="true" />

    <com.google.android.material.chip.Chip
        android:id="@+id/chip2"
        style="@style/Widget.MaterialComponents.Chip.Choice"
        android:minHeight="60dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:chipIcon="@drawable/medium_priority"
        app:iconStartPadding="10dp"
        android:textSize="18dp"
        app:chipIconSize="35dp"
        app:chipIconVisible="true" />

    <com.google.android.material.chip.Chip
        android:id="@+id/chip3"
        android:minHeight="60dp"
        app:checkedIconTint="@color/green"
        style="@style/Widget.MaterialComponents.Chip.Choice"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:chipIcon="@drawable/low_priority"
        android:textSize="18dp"
        app:chipIconSize="35dp"
        app:iconStartPadding="10dp"
        app:chipIconVisible="true" />
</com.google.android.material.chip.ChipGroup>

<Spinner
    android:id="@+id/categorySpinner"
    android:layout_width="match_parent"

```

```
    android:layout_height="wrap_content"
    android:layout_below="@+id/priority"
    android:layout_marginStart="20dp"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="20dp"
    android:layout_marginBottom="20dp" />
```

```
<com.google.android.material.button.MaterialButton
    android:id="@+id/submitBtn"
    style="@style/Widget.MaterialComponents.Button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/categorySpinner"
    android:layout_alignParentEnd="true"
    android:layout_marginHorizontal="20dp"
    android:layout_marginBottom="20dp"
    android:elevation="4dp"
    android:fontFamily="@font/open_sans"
    android:paddingHorizontal="40dp"
    android:paddingVertical="10dp"
    android:text="add task"
```

```
    android:textAppearance="@style/Widget.MaterialComponents.Button.TextButton"
    android:textColor="@color/white"
    android:textSize="18dp"
    app:cornerRadius="25dp" />
```

```
</com.google.android.material.circularreveal.CircularRevealRelativeLayout>
```


Item_recyclerview

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <androidx.cardview.widget.CardView
        android:elevation="20dp"
        android:layout_margin="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <LinearLayout
            android:id="@+id/prio"
            android:background="@drawable/red"
            android:orientation="vertical"
            android:layout_marginHorizontal="5dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:orientation="horizontal">
                <TextView
                    android:id="@+id/textView"
                    style="@style/task_title"
                    android:textColor="@color/black"
                    android:layout_weight="1"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginVertical="5dp"
                    android:layout_marginHorizontal="10dp"
                    android:padding="10dp"
                    android:background="#fff" />
                <!-- <CheckBox
                    android:id="@+id/checkBox"
                    android:layout_weight="8"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_gravity="end"
                    android:onClick="onCheckboxClicked"
                    /> -->
            </LinearLayout>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/hold"
                android:layout_marginHorizontal="10dp"
                android:text=""
                android:textAlignment="textEnd"
                android:paddingBottom="10dp"
                android:paddingHorizontal="10dp"
                android:textColor="@color/primaryDarkColor"
                android:textSize="14dp"
                android:visibility="gone"
            />
        </LinearLayout>
    </CardView>
</LinearLayout>
```

```
        </androidx.cardview.widget.CardView>
    </LinearLayout>
```

Spinner_layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#EEEEEF"
    android:padding="10dp"
    android:orientation="horizontal">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:padding="5dp"
        android:src="@mipmap/ic_launcher" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="18dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:layout_gravity="center"
        android:text="Custom Text"
        android:textColor="#000" />

</LinearLayout>
```