Software for Mobile Devices

Assignment 01

Name: Huraira Muzammal

Section: BCS 6B

Roll number: 22F-3853

Submission on: April 16, 2025

Submitted to: Mam Kanwal Naz

Question 1:

Part I SQL Database

Code

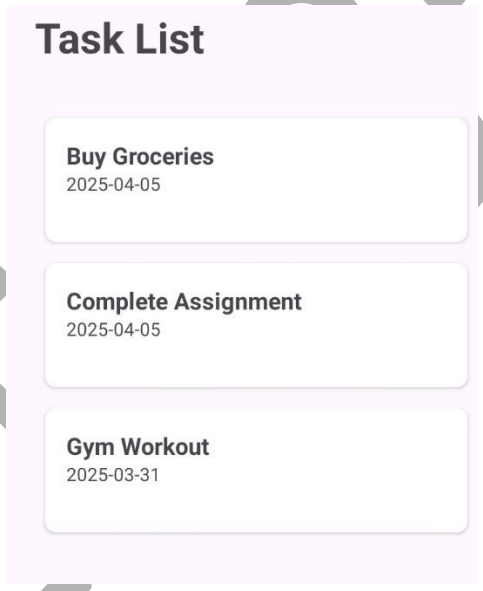
```
package com.example.task_manager;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

import androidx.annotation.Nullable;


public class db_class extends SQLiteOpenHelper {

    private  static final String NAME="MYDATABASE";
    private  static final String TABLE_NAME="taskinfo";
    private static final int VERSION=1;

    private static final String COLUMN_ID = "id";
    private static final String COLUMN_TITLE = "title";
    private static final String COLUMN_DESCRIPTION = "description";
    private static final String COLUMN_DATE = "date";
    private static final String COLUMN_TIME = "time";
    private static final String COLUMN_PRIORITY = "priority";

    public db_class(@Nullable Context context, @Nullable String name, @Nullable
SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        String CREATE_TABLE = "CREATE TABLE " + TABLE_NAME + " (" +
                COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
                COLUMN_TITLE + " TEXT, " +
                COLUMN_DESCRIPTION + " TEXT, " +
```

```
            COLUMN_DATE + " TEXT, " +
            COLUMN_TIME + " TEXT, " +
            COLUMN_PRIORITY + " INTEGER)";
    db.execSQL(CREATE_TABLE);

    // Insert sample tasks if not already present
    insertSampleTasks(db);
}
private void insertSampleTasks(SQLiteDatabase db) {
    // Check if tasks already exist
    Cursor cursor = db.rawQuery("SELECT COUNT(*) FROM " + TABLE_NAME, null);
    cursor.moveToFirst();
    int count = cursor.getInt(0);
    cursor.close();

    // If no tasks exist, insert sample tasks
    if (count == 0) {
        ContentValues values = new ContentValues();

        values.put(COLUMN_TITLE, "Buy Groceries");
        values.put(COLUMN_DESCRIPTION, "Milk, Eggs, Bread");
        values.put(COLUMN_DATE, "2025-04-05");
        values.put(COLUMN_TIME, "8:22 PM");
        values.put(COLUMN_PRIORITY, 1);
        db.insert(TABLE_NAME, null, values);

        values.put(COLUMN_TITLE, "Complete Assignment");
        values.put(COLUMN_DESCRIPTION, "Submit project by Monday");
        values.put(COLUMN_DATE, "2025-04-05");
        values.put(COLUMN_TIME, "8:30 PM");
        values.put(COLUMN_PRIORITY, 2);
        db.insert(TABLE_NAME, null, values);

        values.put(COLUMN_TITLE, "Gym Workout");
        values.put(COLUMN_DESCRIPTION, "Leg day training");
        values.put(COLUMN_DATE, "2025-03-31");
        values.put(COLUMN_TIME, "07:00 AM");
        values.put(COLUMN_PRIORITY, 3);
        db.insert(TABLE_NAME, null, values);

        Log.d("DB_INSERT", "Sample tasks inserted successfully!");
    } else {
        Log.d("DB_INSERT", "Sample tasks already exist. Skipping insertion.");
    }
```

```
}

private boolean hasExistingTasks(SQLiteDatabase db) {
    Cursor cursor = db.rawQuery("SELECT COUNT(*) FROM " + TABLE_NAME, null);
    boolean hasTasks = false;
    if (cursor.moveToFirst()) {
        hasTasks = cursor.getInt(0) > 0; // If count > 0, tasks exist
    }
    cursor.close();
    return hasTasks;
}

public long addTask(String title, String description, String date, String time, int priority) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(COLUMN_TITLE, title);
    values.put(COLUMN_DESCRIPTION, description);
    values.put(COLUMN_DATE, date);
    values.put(COLUMN_TIME, time);
    values.put(COLUMN_PRIORITY, priority);

    long id = db.insert(TABLE_NAME, null, values);
    db.close();
    return id;
}
public Cursor getAllTasks() {
    SQLiteDatabase db = this.getReadableDatabase();
    return db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
}

public void updateTask(int id, String title, String description, String date, String time, int priority) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(COLUMN_TITLE, title);
    values.put(COLUMN_DESCRIPTION, description);
    values.put(COLUMN_DATE, date);
    values.put(COLUMN_TIME, time);
    values.put(COLUMN_PRIORITY, priority);

    db.update(TABLE_NAME, values, COLUMN_ID + "=?", new String[]{String.valueOf(id)});
    db.close();
}

public void deleteTask(int id) {
```

```
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_NAME, COLUMN_ID + "=?", new String[]{String.valueOf(id)});
        db.close();
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
}
```

```
2025-04-13 01:47:57.214 31591-31591 DB_INSERT            com.example.task_manager    D  Sample tasks inserted successfully!
2025-04-13 01:47:57.228 31591-31591 DB_CHECK             com.example.task_manager    D  Database opened successfully!
```



ii. Services

```
package com.example.task_manager;

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.IBinder;
import android.provider.Settings;

import androidx.annotation.Nullable;

public class Notification extends Service {
```

```
    MediaPlayer ringtone;
    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        ringtone = MediaPlayer.create(this,
                Settings.System.DEFAULT_NOTIFICATION_URI);
//        ringtone.setLooping(true); // Loop the ringtone indefinitely
        ringtone.start(); // Start the ringtone
        return START_NOT_STICKY;
    }

    @Override
    public void onDestroy() {
        ringtone.stop(); // Stop the ringtone
        super.onDestroy(); // Call superclass method for cleanup
    }
}
```

iii. Permissions

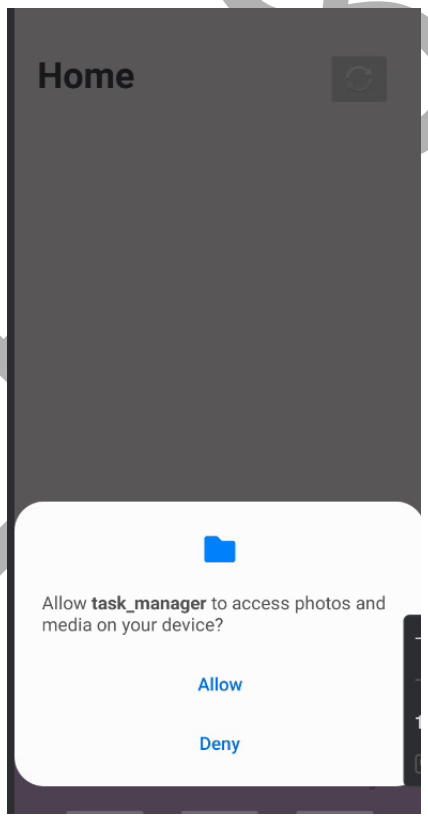
```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
    android:name="android.permission.MANAGE_EXTERNAL_STORAGE"
    tools:ignore="ScopedStorage" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Main file

```
private void requestStoragePermission() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {  // Android 11+
        if (!android.os.Environment.isExternalStorageManager()) {
            // Show permission request toast
            Toast.makeText(this, "Requesting Manage Storage Permission", Toast.LENGTH_SHORT).show();

            // Open settings to grant permission
            Intent intent = new Intent(Settings.ACTION_MANAGE_APP_ALL_FILES_ACCESS_PERMISSION);
```

```
            intent.setData(Uri.parse("package:" + getPackageName()));
            startActivityForResult(intent, MANAGE_STORAGE_PERMISSION_REQUEST_CODE);
        } else {
            Toast.makeText(this, "Manage Storage Permission already granted",
Toast.LENGTH_SHORT).show();
        }
    } else { // Android 10 and below
        if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED ||
                ContextCompat.checkSelfPermission(this, Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {

            ActivityCompat.requestPermissions(this, new String[]{
                    Manifest.permission.READ_EXTERNAL_STORAGE,
                    Manifest.permission.WRITE_EXTERNAL_STORAGE
            }, STORAGE_PERMISSION_REQUEST_CODE);

            Toast.makeText(this, "Requesting storage permissions", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(this, "Storage permission already granted", Toast.LENGTH_SHORT).show();
        }
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull
int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    if (requestCode == STORAGE_PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0) {
            boolean readGranted = grantResults[0] == PackageManager.PERMISSION_GRANTED;
            boolean writeGranted = grantResults.length > 1 && grantResults[1] ==
PackageManager.PERMISSION_GRANTED;

            if (readGranted && writeGranted) {
                Toast.makeText(this, "Storage permission granted", Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "Storage permission denied", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == MANAGE_STORAGE_PERMISSION_REQUEST_CODE) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {
            if (Environment.isExternalStorageManager()) {
                Toast.makeText(this, "Manage Storage Permission granted", Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "Manage Storage Permission denied", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

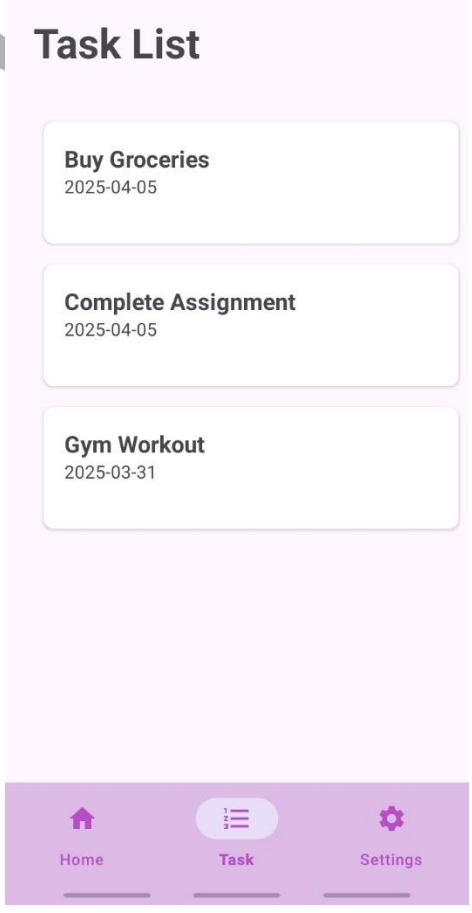


iv. RecyclerView & CardView

task list fragment.java

```
package com.example.task_manager;

import android.database.Cursor;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;
import java.util.List;
public class TaskListFragment extends Fragment {
    private RecyclerView recyclerView;
    private TaskAdapter taskAdapter;
    private List<TaskModel> taskList;
    private db_class db;

    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_task_list, container, false);

        recyclerView = view.findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));

        db = new db_class(getContext(), "MYDATABASE", null, 1);
        loadTasks();

        return view;
    }

    private void loadTasks() {
        taskList = new ArrayList<>();
        Cursor cursor = db.getAllTasks();

        if (cursor != null && cursor.moveToFirst()) {
            do {
                int id = cursor.getInt(cursor.getColumnIndexOrThrow("id"));  // Fix
                String title = cursor.getString(cursor.getColumnIndexOrThrow("title"));  // Fix
```

```
            String description = cursor.getString(cursor.getColumnIndexOrThrow("description"));
            String date = cursor.getString(cursor.getColumnIndexOrThrow("date"));
            String time = cursor.getString(cursor.getColumnIndexOrThrow("time"));
            int priority = cursor.getInt(cursor.getColumnIndexOrThrow("priority"));
            TaskModel task = new TaskModel(id, title, description, date, time, priority);
            taskList.add(task);
        } while (cursor.moveToNext());
        cursor.close();
    }

    taskAdapter = new TaskAdapter(getContext(), taskList, task -> openTaskDetails(task));
    recyclerView.setAdapter(taskAdapter);
  }

  private void openTaskDetails(TaskModel task) {
    TaskDetailsFragment taskDetailsFragment = new TaskDetailsFragment();
    Bundle bundle = new Bundle();
    bundle.putInt("task_id", task.getId());
    bundle.putString("task_title", task.getTitle());
    bundle.putString("task_description",
task.getDescription());
    bundle.putString("task_date", task.getDate());
    bundle.putString("task_time", task.getTime());
    bundle.putInt("task_priority", task.getPriority());
    taskDetailsFragment.setArguments(bundle);

getActivity().getSupportFragmentManager().beginTransaction()
            .replace(R.id.frameLayout, taskDetailsFragment)
            .addToBackStack(null)
            .commit();
  }
}
```



v. Fragments (Modular UI Design)

settingfragment

```
package com.example.task_manager;

import android.annotation.SuppressLint;
import android.content.SharedPreferences;
import android.os.Bundle;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Switch;

public class SettingsFragment extends Fragment {

    private SharedPreferences sharedPreferences;
    private static final String PREF_NAME = "settings_prefs";
    private static final String KEY_NOTIFICATIONS = "notifications_enabled";

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_setting2, container, false);
    }

    @Override
    public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        // Initialize the Switch
        @SuppressLint("UseSwitchCompatOrMaterialCode") Switch switchNotifications =
view.findViewById(R.id.notification_switch);

        // Initialize SharedPreferences
        sharedPreferences = getActivity().getSharedPreferences(PREF_NAME, getActivity().MODE_PRIVATE);

        // Load the saved state
        boolean isChecked = sharedPreferences.getBoolean(KEY_NOTIFICATIONS, false);
        switchNotifications.setChecked(isChecked);

        // Save state when toggled
```

```
        switchNotifications.setOnCheckedChangeListener((buttonView, isChecked1) -> {
            SharedPreferences.Editor editor = sharedPreferences.edit();
            editor.putBoolean(KEY_NOTIFICATIONS, isChecked1);
            editor.apply(); // Save changes
        });
    }

}
```

homefragment

```
package com.example.task_manager;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

import androidx.fragment.app.Fragment;

import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

public class HomeFragment extends Fragment {

    private ImageView imageView;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_home, container, false);

        imageView = view.findViewById(R.id.quoteImageView);

        // Load the image in background thread
        loadImage();

        return view;
    }
```

```
    private void loadImage() {
        ExecutorService executor = Executors.newSingleThreadExecutor();
        executor.execute(() -> {
            try {
                URL url = new URL("https://muslimmemo.com/wp-content/uploads/2016/06/islamic-quote-
umar-patience.png");
                HttpURLConnection connection = (HttpURLConnection) url.openConnection();
                connection.setDoInput(true);
                connection.connect();
                InputStream input = connection.getInputStream();
                Bitmap bitmap = BitmapFactory.decodeStream(input);

                // Update UI on main thread
                requireActivity().runOnUiThread(() -> imageView.setImageBitmap(bitmap));

            } catch (Exception e) {
                e.printStackTrace();
            }
        });
    }
}
```

taskfragment

```
package com.example.task_manager;

import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class TaskFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_task, container, false);
    }
}
```

task detail fragment

```
package com.example.task_manager;

import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import java.io.Serializable;
public class TaskDetailsFragment extends Fragment {
   private TextView titleTextView, descriptionTextView, dateTextView, timeTextView, priorityTextView;

   public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
      View view = inflater.inflate(R.layout.fragment_task_details, container, false);

      titleTextView = view.findViewById(R.id.taskTitle);
      descriptionTextView = view.findViewById(R.id.taskDescription);
      dateTextView = view.findViewById(R.id.taskDueDate);
      timeTextView = view.findViewById(R.id.taskTime);
      priorityTextView = view.findViewById(R.id.taskPriority);

      Bundle bundle = getArguments();
      if (bundle != null) {
         titleTextView.setText(bundle.getString("task_title"));
         descriptionTextView.setText(bundle.getString("task_description"));
         dateTextView.setText(bundle.getString("task_date"));
         timeTextView.setText(bundle.getString("task_time"));

         int priority = bundle.getInt("task_priority");
         priorityTextView.setText("Priority: " + priority);
      }

      return view;
   }
}
```

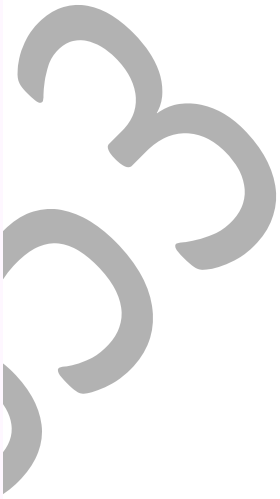
Buy Groceries

Milk, Eggs, Bread

2025-04-05

8:22 PM

Priority: 1



Home Task Settings

iii. Shared Preferences

```
sharedPreferences = getActivity().getSharedPreferences(PREF_NAME, getActivity().MODE_PRIVATE);

// Load the saved state
boolean isChecked = sharedPreferences.getBoolean(KEY_NOTIFICATIONS, false);
switchNotifications.setChecked(isChecked);

// Save state when toggled
switchNotifications.setOnCheckedChangeListener((buttonView, isChecked1) -> {
   SharedPreferences.Editor editor = sharedPreferences.edit();
   editor.putBoolean(KEY_NOTIFICATIONS, isChecked1);
   editor.apply(); // Save changes
});
```

iii. Internet call

```
private void loadImage() {
   ExecutorService executor = Executors.newSingleThreadExecutor();
   executor.execute(() -> {
      try {
         URL url = new URL("https://muslimmemo.com/wp-content/uploads/2016/06/islamic-quote-
umar-patience.png");
         HttpURLConnection connection = (HttpURLConnection) url.openConnection();
         connection.setDoInput(true);
         connection.connect();
         InputStream input = connection.getInputStream();
         Bitmap bitmap = BitmapFactory.decodeStream(input);

         // Update UI on main thread
         requireActivity().runOnUiThread(() -> imageView.setImageBitmap(bitmap));

      } catch (Exception e) {
         e.printStackTrace();
      }
   });
}
```

iii. Navigation Bar

xml code

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/nav_home"
        android:icon="@drawable/home"
        android:title="Home" />

    <item
        android:id="@+id/nav_task"
        android:icon="@drawable/list"
        android:title="Task" />

    <item
        android:id="@+id/nav_settings"
        android:icon="@drawable/setting"
        android:title="Settings" />
```

```
</menu>
```

Activity main

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <!-- FrameLayout to hold fragments -->
    <FrameLayout
        android:id="@+id/frameLayout"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"/>

    <!-- Bottom Navigation Bar -->
    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottomNavigationView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:menu="@menu/bottom_nav_menu"
        android:background="@color/nav_item_color"
        app:itemIconTint="@color/nav_item_color_selected"
        app:itemTextColor="@color/nav_item_color_selected"
        app:itemRippleColor="@color/nav_item_color_selected"
        app:elevation="8dp"/>


</LinearLayout>
```

Bottom nav bar java

```
package com.example.task_manager;

import android.os.Bundle;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
```

```
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class bottom_nav_bar extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_bottom_nav_bar);

    }
}
```

End of code file