**Software for Mobile Devices**

**Assignment 01**

Name: Huraira Muzammal

Section: BCS 6B

Roll number: 22F-3853

Submission on: April 16, 2025

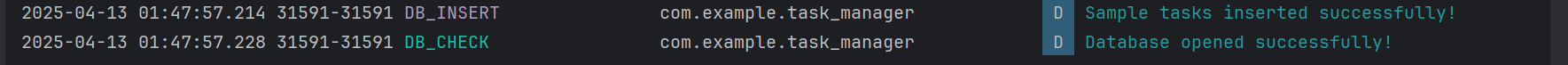
Submitted to: Mam Kanwal Naz

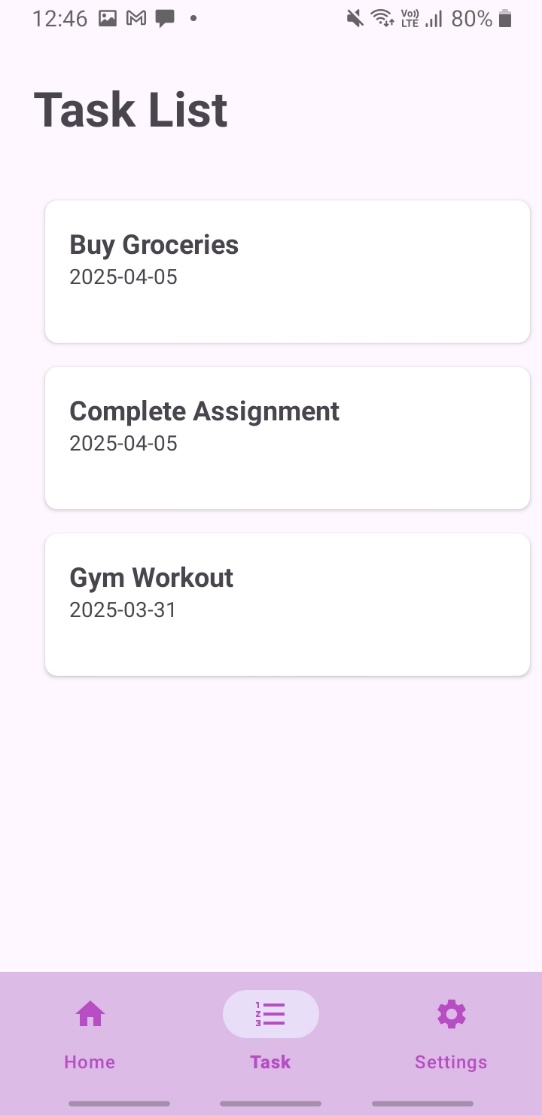
Question 1:

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);  
 }  
}





ii. Services

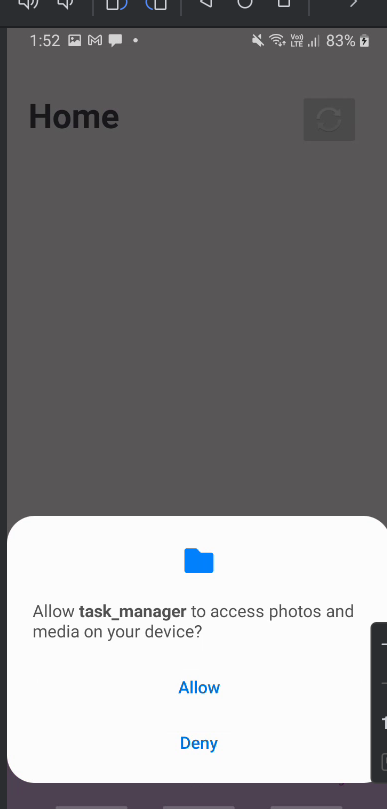
package com.example.task\_manager;  
  
import android.app.Service;  
import android.content.Intent;  
import android.media.MediaPlayer;  
import android.os.IBinder;  
import android.provider.Settings;  
  
import androidx.annotation.Nullable;  
  
public class Notification extends Service {  
  
  
 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());  
A screenshot of a cell phone

AI-generated content may be incorrect. 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

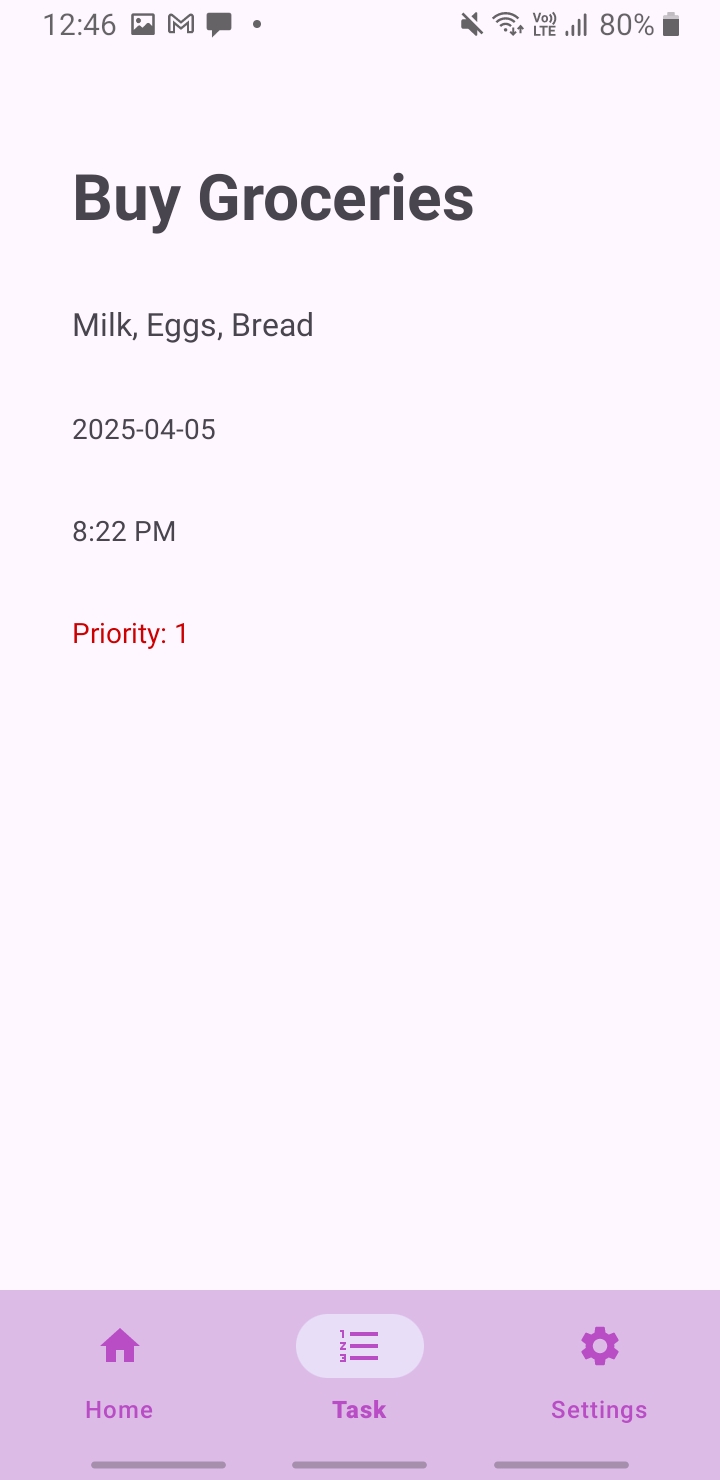
package com.example.task\_manager;  
  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
  
import androidx.fragment.app.Fragment;  
  
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;  
 }  
}

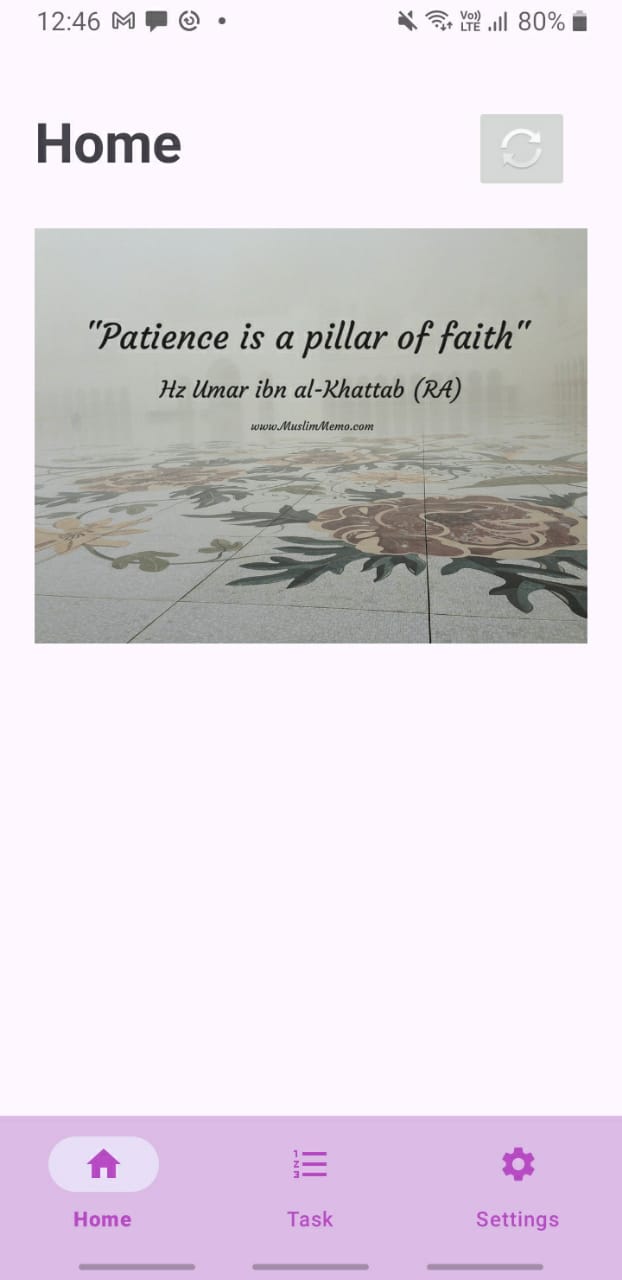


iii. Shared Preferences

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*);  
  
 }  
}

A close-up of a sign

AI-generated content may be incorrect.

End of code file