
# ST AGNES COLLEGE (AUTONOMOUS), MANGALURU

RE-ACCREDITED BY NAAC WITH A+ GRADE CGPA 3.65/4(CYCLE 4)

College of Excellence by UGC, New Delhi

Star College Status by DBT, MST, Govt. of India

**Project on**

## "ROOM DATABASE"

## MASTER OF COMPUTER APPLICATION

## (MCA)

**Team Members:**

**Name: Aarol Royston Lobo** Roll No:2384059

**Name: Clevin D'silva** Roll No:2384066

**Name: Denson Brian Nazareth** Roll No:2384092

**Name:Rachan** Roll No:2384105

Under the esteemed Guidance of

**Mrs. Baji Raina Banu**

Staff Co-ordinator Dept of Computer Application

**Submitted on: 10/09/2024**

**Subject: Mobile Application Development**

**Signature of Faculty incharge:_____**

**activity_main.xml**

```
<?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="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <!-- ListView to display note titles -->

    <!-- EditText to enter note title -->

    <EditText
        android:id="@+id/editTextTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Title"
        android:minHeight="48dp" />

    <!-- EditText to enter note content -->
    <EditText
        android:id="@+id/editTextContent"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Content"
        android:minHeight="48dp" />

    <!-- Buttons for actions -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_horizontal">

        <Button
            android:id="@+id/buttonAdd"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Add" />

        <Button
            android:id="@+id/buttonUpdate"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Update" />
```

```
        <Button
            android:id="@+id/buttonDelete"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Delete" />
    </LinearLayout>
    <ListView
        android:id="@+id/listViewNotes"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:divider="@android:color/darker_gray"
        android:dividerHeight="1dp" />

</LinearLayout>
```

**MainActivity.kt**

```
package com.denson.noteapp

import android.os.Bundle
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
import androidx.lifecycle.ViewModelProvider
import androidx.lifecycle.Observer

class MainActivity : AppCompatActivity() {

    private lateinit var listViewNotes: ListView
    private lateinit var editTextTitle: EditText
    private lateinit var editTextContent: EditText
    private lateinit var buttonAdd: Button
    private lateinit var buttonUpdate: Button
    private lateinit var buttonDelete: Button

    private lateinit var noteViewModel: NoteViewModel
    private var selectedNote: Note? = null

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

        listViewNotes = findViewById(R.id.listViewNotes)
        editTextTitle = findViewById(R.id.editTextTitle)
        editTextContent = findViewById(R.id.editTextContent)
        buttonAdd = findViewById(R.id.buttonAdd)
```

```
buttonUpdate = findViewById(R.id.buttonUpdate)
buttonDelete = findViewById(R.id.buttonDelete)

// Initialize ViewModel
noteViewModel = ViewModelProvider(this).get(NoteViewModel::class.java)

// Observe LiveData for changes
noteViewModel.allNotes.observe(this, Observer { notes ->
    // Create a simple adapter to display note titles
    val titles = notes.map { it.title }
    val adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1, titles)
    listViewNotes.adapter = adapter
})
// Handle Add Button
buttonAdd.setOnClickListener {
    val title = editTextTitle.text.toString()
    val content = editTextContent.text.toString()
    if (title.isNotEmpty() && content.isNotEmpty()) {
        val note = Note(title = title, content = content)
        noteViewModel.insert(note)
        clearInputs()
    }
}
// Handle Update Button
buttonUpdate.setOnClickListener {
    val title = editTextTitle.text.toString()
    val content = editTextContent.text.toString()
    selectedNote?.let {
        if (title.isNotEmpty() && content.isNotEmpty()) {
            val updatedNote = it.copy(title = title, content = content)
            noteViewModel.update(updatedNote)
            clearInputs()
        }
    }
}
// Handle Delete Button
buttonDelete.setOnClickListener {
    selectedNote?.let {
        noteViewModel.delete(it.id)
        clearInputs()
    }
}
// Handle ListView Item Click
listViewNotes.setOnItemClickListener { _, _, position, _ ->
    val notes = noteViewModel.allNotes.value ?: return@setOnItemClickListener
    selectedNote = notes[position]
```

```
            editTextTitle.setText(selectedNote?.title)
            editTextContent.setText(selectedNote?.content)
        }
    }
    // Helper function to clear input fields
    private fun clearInputs() {
        editTextTitle.text.clear()
        editTextContent.text.clear()
        selectedNote = null
    }
}
```

**Note.kt**

```
package com.denson.noteapp

import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity(tableName = "notes")
data class Note(
    @PrimaryKey(autoGenerate = true) val id: Int = 0,
    val title: String,
    val content: String
)
```

**NoteDao.kt**

```
package com.denson.noteapp

import androidx.lifecycle.LiveData
import androidx.room.Dao
import androidx.room.Insert
import androidx.room.Query
import androidx.room.Update

@Dao
interface NoteDao {
    @Insert
    suspend fun insert(note: Note)

    @Update
    suspend fun update(note: Note)

    @Query("SELECT * FROM notes")
```

```
    fun getAllNotes(): LiveData<List<Note>>

    @Query("DELETE FROM notes WHERE id = :noteId")
    suspend fun delete(noteId: Int)
}
```

**NoteDatabase.kt**

```
package com.denson.noteapp

import  android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase

@Database(entities = [Note::class], version = 1)
abstract class NoteDatabase : RoomDatabase() {
    abstract fun noteDao(): NoteDao

    companion object {
        @Volatile
        private var INSTANCE: NoteDatabase? = null

        fun getDatabase(context: Context): NoteDatabase {
            return INSTANCE ?: synchronized(this) {
                val instance = Room.databaseBuilder(
                    context.applicationContext,
                    NoteDatabase::class.java,
                    "note_database"
                ).build()
                INSTANCE = instance
                instance
            }
        }
    }
}
```

**NoteRepository.kt**

```
package com.denson.noteapp

import androidx.lifecycle.LiveData

class NoteRepository(private val noteDao: NoteDao) {

    val allNotes: LiveData<List<Note>> = noteDao.getAllNotes()

    suspend fun insert(note: Note) {
```

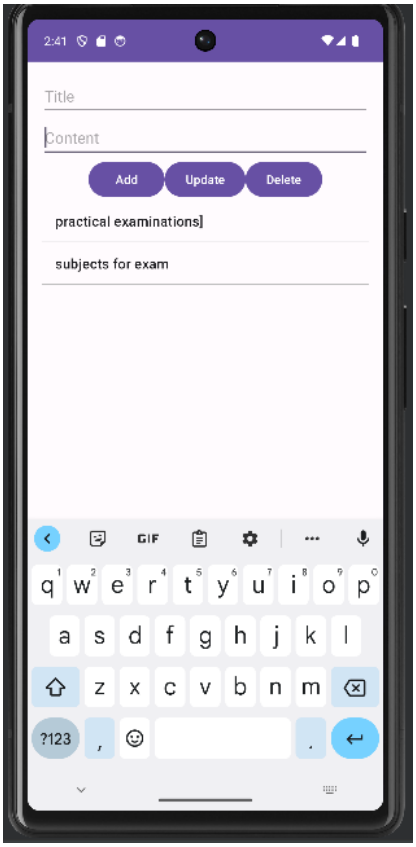
```
        noteDao.insert(note)
    }

    suspend fun update(note: Note) {
        noteDao.update(note)
    }

    suspend fun delete(noteId: Int) {
        noteDao.delete(noteId)
    }
}
```

**NoteViewModel.kt**

```
package com.denson.noteapp

import android.app.Application
import androidx.lifecycle.AndroidViewModel
import androidx.lifecycle.LiveData
import androidx.lifecycle.viewModelScope
import kotlinx.coroutines.launch

class NoteViewModel(application: Application) : AndroidViewModel(application) {

    private val repository: NoteRepository
    val allNotes: LiveData<List<Note>>

    init {
        val noteDao = NoteDatabase.getDatabase(application).noteDao()
        repository = NoteRepository(noteDao)
        allNotes = repository.allNotes
    }

    fun insert(note: Note) = viewModelScope.launch {
        repository.insert(note)
    }

    fun update(note: Note) = viewModelScope.launch {
        repository.update(note)
    }

    fun delete(noteId: Int) = viewModelScope.launch {
        repository.delete(noteId)
    }
}
```

**Output:**


2:43
Mon, Sep 9
Gmail
Photos
YouTube


2:41
Title
Content
Add
Update
Delete
practical examinations]
subjects for exam
GIF
?123