



PROJECT REPORT ON

My Notes

(UNDER THE PARTIAL FULFILMENT OF B.Sc. DEGREE COURSE IN COMPUTER SCIENCE)

SUBMITTED BY:

RANA SUJEET KUMAR

GUIDED BY:

Dr. Mrs. SALONI BHUSHAN

PRO. Mr. RAJESH YADAV

DEPARTMENT OF COMPUTER SCIENCE
V.K. KRISHNA MENON COLLEGE OF COMMERCE AND ECONOMICS
&
SHARAD SHANKAR DIGHE COLLEGE OF SCIENCE
BHANDUP (EAST), MUMBAI-400042

UNIVERSITY OF MUMBAI

Year 2021-22

CERTIFICATE

**V.K KRISHNA MENON COLLEGE OF COMMERCE
AND ECONOMICS
&
SHARAD SHANKAR DIGHE COLLEGE OF
SCIENCE BHANDUP EAST, MUMBAI-400042**

This is to certify that Mr. /Miss **RANA SUJEET KUMAR MAHESH**
Seat No. **38** has successfully completed **Project Implementation** Practical's
of **Semester-VI** for partial fulfilment of **B.Sc. Degree** course in **Computer
Science** of University of Mumbai in academic year **2021-2022** under the
guidance of **Dr. Saloni Bhushan & Rajesh Yadav**

**DATE:
DEPARTMENT**

HEAD OF THE

Teacher-in-charge

EXAMINER

INDEX

| Sr.No. | Title | Page No. |
|---------------|---------------------------|-----------------|
| 1. | Acknowledgement | 5 |
| 2. | Abstract | 5 |
| 3. | Undertaking | 6 |
| 4. | Introduction | 7 |
| 5. | Requirement Specification | 7 |
| 6. | Gantt Chart | 8 |
| 7. | E-R Diagram | 9 |
| 8. | Code Implementation | 10 |
| 9. | Results | 47 |
| 10. | Reference | 49 |

ACKNOWLEDGEMENT

It gives great pleasure and pride as I presented my project on "My Notes " This acknowledgement is a small effort to express and our gratitude to all those who have shown me the path to bring out the various colors of this project with their vast treasure of experience and knowledge. I would like to express my sincere thanks to Prof. Mr. Rajesh Yadav, who helps me throughout the project for providing me moral support, conducive work environment and the much-needed inspiration to conclude this project on time. I also take this opportunity to thank Head of the Department Dr. Saloni Bhushan and thanks to all of our professors of the Department of Computer Science of V. K. Krishna Menon College for giving me an opportunity to study in the institute. He was the Source of continuous encouragement as milestone was crossed. Thank You so much.

Abstract

- Lets Note is an application that provide all the users to make notes , list
- Using this application the user make notes and give the colour hint for memorize the notes
- This application your notes data save in your mobile
- Users easily install this app and make notes
- Users can also upload photos and links

UNDERTAKING

This is to declare that the project titled "**My Notes**" is an original work done by the undersigned, in partial fulfilment of the requirement for the degree "Bachelor of Science in Computer Science" at CS Department, V. K. Krishna Menon College. All the analysis design and System developed have been accomplished by the undersigned. Moreover, this project hasnot been submitted to any other College or University.

Mr. Rana Sujeet Kumar Mahesh

Introduction

- This application is for those person who use application without any security threat
- Customer gets the freedom to use this app access all the features free

Requirement Specification

- **Software Requirements**

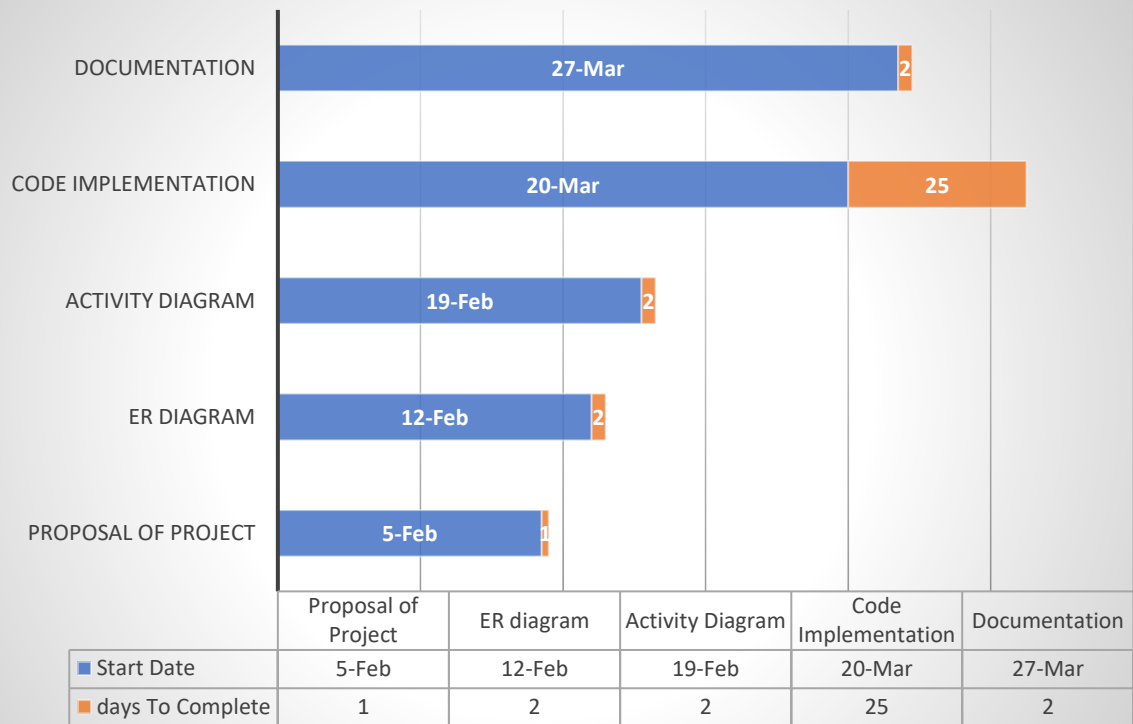
Android studio

Open SDK

- **Hardware requirement:**

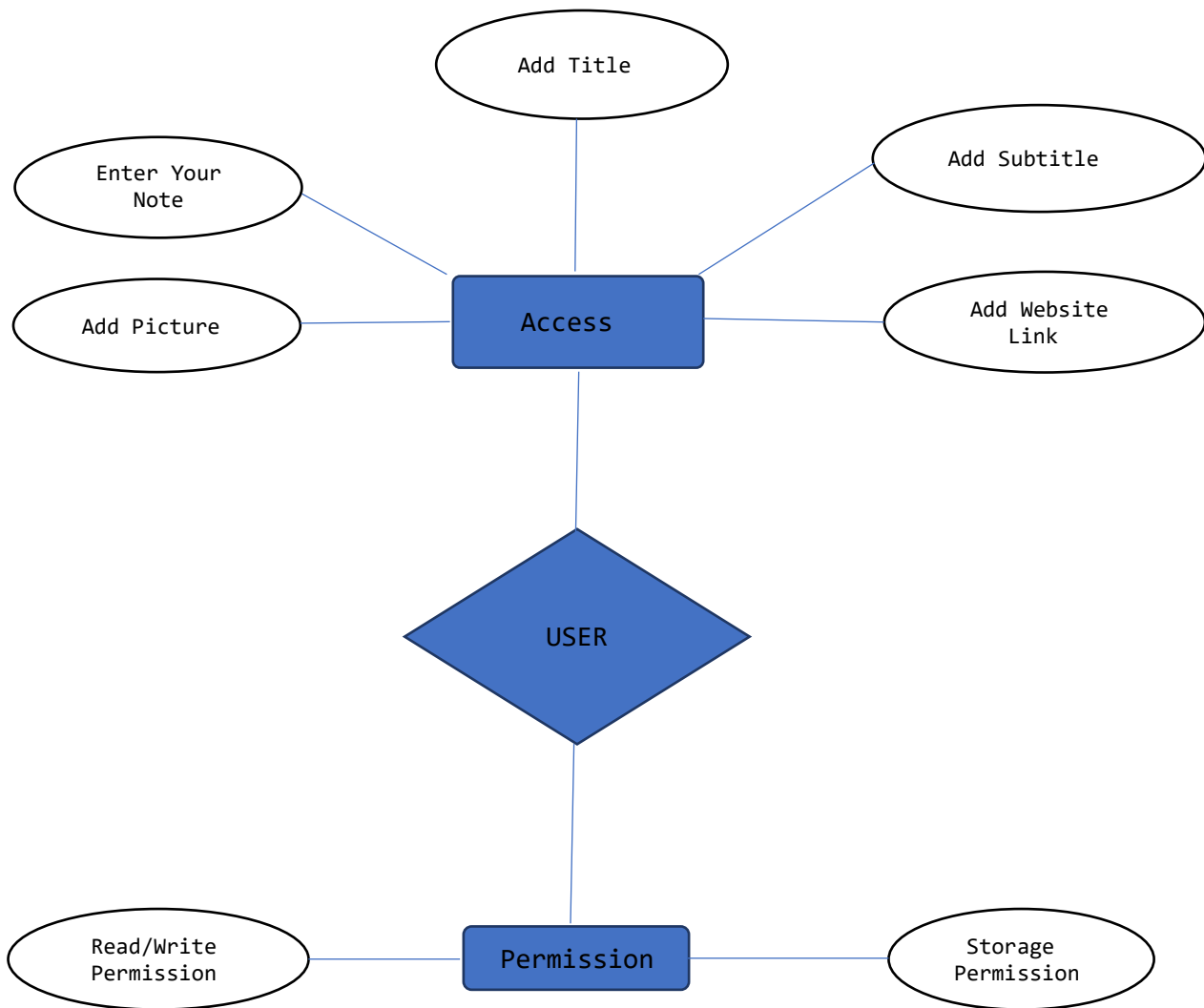
| | | |
|-----------|---|------|
| Processor | : | I5 |
| RAM | : | 8GB |
| Hard Disk | : | 50GB |

Gantt Chart



System design details

ER Diagram:



Code Implementation

MainActivity.kt

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.fragment.app.Fragment
import androidx.fragment.app.FragmentManager

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

        replaceFragment(HomeFragment.newInstance(), false)
    }

    fun replaceFragment(fragment: Fragment, istransition: Boolean) {
        val fragmentTransition = supportFragmentManager.beginTransaction()

        if (istransition) {
            fragmentTransition.setCustomAnimations(android.R.anim.slide_out_right, android.R.anim.slide_in_left)
        }

        fragmentTransition.add(R.id.frame_layout, fragment).addToBackStack(fragment.javaClass.simpleName).commit()
    }

    override fun onBackPressed() {
        super.onBackPressed()
        val fragments = supportFragmentManager.fragments
        if (fragments.size == 0) {
            finish()
        }
    }
}

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.fragment.app.Fragment
import androidx.fragment.app.FragmentManager

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

        replaceFragment(HomeFragment.newInstance(), false)
```

```

    }

    fun replaceFragment(fragment: Fragment, istransition: Boolean) {
        val fragmentTransition = supportFragmentManager.beginTransaction()

        if (istransition) {
            fragmentTransition.setCustomAnimations(android.R.anim.slide_out_right, android.R.anim.slide_in_left)
        }

        fragmentTransition.add(R.id.frame_layout, fragment).addToBackStack(fragment.javaClass.simpleName).commit()
    }

    override fun onBackPressed() {
        super.onBackPressed()
        val fragments = supportFragmentManager.fragments
        if (fragments.size == 0) {
            finish()
        }
    }
}

```

NotesAdapter.kt

```

import android.graphics.BitmapFactory
import android.graphics.Color
import android.graphics.drawable.GradientDrawable
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.codingwithme.notesapp.R
import com.codingwithme.notesapp.entities.Notes
import kotlinx.android.synthetic.main.item_rv_notes.view.*
import java.util.*

class NotesAdapter() :
    RecyclerView.Adapter<NotesAdapter.NotesViewHolder>() {
    var listener: OnItemClickListener? = null
    var arrList = ArrayList<Notes>()
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
        NotesViewHolder {
        return NotesViewHolder(
            LayoutInflater.from(parent.context).inflate(R.layout.item_rv_notes, parent, false)
        )
    }

    override fun getItemCount(): Int {
        return arrList.size
    }
}

```

```
}

fun setData(arrNotesList: List<Notes>){
    arrList = arrNotesList as ArrayList<Notes>
}

fun setOnClickListener(listener1: OnItemClickListener){
    listener = listener1
}

override fun onBindViewHolder(holder: NotesViewHolder, position: Int) {

    holder.itemView.tvTitle.text = arrList[position].title
    holder.itemView.tvDesc.text = arrList[position].noteText
    holder.itemView.tvDateTime.text = arrList[position].dateTime

    if (arrList[position].color != null){
holder.itemView.cardView.setCardBackgroundColor(Color.parseColor(arrList[posit
ion].color))
    }else{
holder.itemView.cardView.setCardBackgroundColor(Color.parseColor(R.color.Colo
rLightBlack.toString()))
    }

    if (arrList[position].imgPath != null){
holder.itemView.imgNote.setImageBitmap(BitmapFactory.decodeFile(arrList[posit
ion].imgPath))
        holder.itemView.imgNote.visibility = View.VISIBLE
    }else{
        holder.itemView.imgNote.visibility = View.GONE
    }

    if (arrList[position].webLink != ""){
        holder.itemView.tvWebLink.text = arrList[position].webLink
        holder.itemView.tvWebLink.visibility = View.VISIBLE
    }else{
        holder.itemView.tvWebLink.visibility = View.GONE
    }

    holder.itemView.cardView.setOnClickListener {
        listener!!.onClicked(arrList[position].id!!)
    }

}

class NotesViewHolder(view:View) : RecyclerView.ViewHolder(view){

}

interface OnItemClickListener{
    fun onClicked(noteId:Int)
}

}
```

NoteDao.kt

```
import androidx.room.*
import com.codingwithme.notesapp.entities.Notes

@Dao
interface NoteDao {

    @Query("SELECT * FROM notes ORDER BY id DESC")
    suspend fun getAllNotes() : List<Notes>

    @Query("SELECT * FROM notes WHERE id =:id")
    suspend fun getSpecificNote(id:Int) : Notes

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insertNotes(note:Notes)

    @Delete
    suspend fun deleteNote(note:Notes)

    @Query("DELETE FROM notes WHERE id =:id")
    suspend fun deleteSpecificNote(id:Int)

    @Update
    suspend fun updateNote(note:Notes)
}
```

NotesDatabase.kt

```
import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase
import com.codingwithme.notesapp.dao.NoteDao
import com.codingwithme.notesapp.entities.Notes

@Database(entities = [Notes::class], version = 1, exportSchema = false)
abstract class NotesDatabase : RoomDatabase() {

    companion object {
        var notesDatabase: NotesDatabase? = null

        @Synchronized
        fun getDatabase(context: Context): NotesDatabase {
            if (notesDatabase == null) {
                notesDatabase = Room.databaseBuilder(
                    context,
                    NotesDatabase::class.java,
                    "notes.db"
                ).build()
            }
            return notesDatabase!!
        }
    }
}
```

```
}  
  
    abstract fun noteDao():NoteDao  
}
```

Notes.kt

```
import androidx.room.ColumnInfo  
import androidx.room.Entity  
import androidx.room.PrimaryKey  
import java.io.Serializable  
  
@Entity(tableName = "Notes")  
class Notes: Serializable {  
  
    @PrimaryKey(autoGenerate = true)  
    var id:Int? = null  
  
    @ColumnInfo(name = "title")  
    var title:String? = null  
  
    @ColumnInfo(name = "sub_title")  
    var subTitle:String? = null  
  
    @ColumnInfo(name = "date_time")  
    var dateTime:String? = null  
  
    @ColumnInfo(name = "note_text")  
    var noteText:String? = null  
  
    @ColumnInfo(name = "img_path")  
    var imgPath:String? = null  
  
    @ColumnInfo(name = "web_link")  
    var webLink:String? = null  
  
    @ColumnInfo(name = "color")  
    var color:String? = null  
  
    override fun toString(): String {  
        return "$title : $dateTime"  
    }  
}
```

NoteBottomsheetfragment

```
import android.app.Dialog
import android.content.Intent
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.coordinatorlayout.widget.CoordinatorLayout
import androidx.localbroadcastmanager.content.LocalBroadcastManager
import com.codingwithme.notesapp.R
import com.google.android.material.bottomsheet.BottomSheetBehavior
import com.google.android.material.bottomsheet.BottomSheetDialogFragment
import kotlinx.android.synthetic.main.fragment_notes_bottom_sheet.*

class NoteBottomSheetFragment : BottomSheetDialogFragment() {
    var selectedColor = "#171C26"

    companion object {
        var noteId = -1
        fun newInstance(id: Int): NoteBottomSheetFragment {
            val args = Bundle()
            val fragment = NoteBottomSheetFragment()
            fragment.arguments = args
            noteId = id
            return fragment
        }
    }

    override fun setupDialog(dialog: Dialog, style: Int) {
        super.setupDialog(dialog, style)

        val view =
            LayoutInflater.from(context).inflate(R.layout.fragment_notes_bottom_sheet, null)
        dialog.setContentView(view)

        val param = (view.parent as View).layoutParams as
            CoordinatorLayout.LayoutParams

        val behavior = param.behavior

        if (behavior is BottomSheetBehavior<*>){
            behavior.setBottomSheetCallback(object :
                BottomSheetBehavior.BottomSheetCallback() {
                override fun onSlide(bottomSheet: View, slideOffset: Float) {
                    TODO("Not yet implemented")
                }

                override fun onStateChanged(bottomSheet: View, newState: Int)
            {
                var state = ""
                when (newState) {
                    BottomSheetBehavior.STATE_DRAGGING -> {
                        state = "DRAGGING"
                    }
                }
            }
        }
    }
}
```

```

        }
        BottomSheetBehavior.STATE_SETTLING -> {
            state = "SETTLING"
        }
        BottomSheetBehavior.STATE_EXPANDED -> {
            state = "EXPANDED"
        }
        BottomSheetBehavior.STATE_COLLAPSED -> {
            state = "COLLAPSED"
        }

        BottomSheetBehavior.STATE_HIDDEN -> {
            state = "HIDDEN"
            dismiss()
            behavior.state =
BottomSheetBehavior.STATE_COLLAPSED
        }
    }
}

}))

}

}

override fun onCreateView(
    inflater: LayoutInflater,
    container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    return
inflater.inflate(R.layout.fragment_notes_bottom_sheet, container, false)

}

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    if (noteId != -1){
        layoutDeleteNote.visibility = View.VISIBLE
    }else{
        layoutDeleteNote.visibility = View.GONE
    }
    setListener()
}

private fun setListener(){
    fNote1.setOnClickListener {

        imgNote1.setImageResource(R.drawable.ic_tick)
        imgNote2.setImageResource(0)
        imgNote4.setImageResource(0)
        imgNote5.setImageResource(0)
        imgNote6.setImageResource(0)
        imgNote7.setImageResource(0)
        selectedColor = "#4e33ff"
    }
}

```



```
        val intent = Intent("bottom_sheet_action")
        intent.putExtra("action","Blue")
        intent.putExtra("selectedColor",selectedColor)

LocalBroadcastManager.getInstance(requireContext()).sendBroadcast(intent)

    }

    fNote2.setOnClickListener {
        imgNote1.setImageResource(0)
        imgNote2.setImageResource(R.drawable.ic_tick)
        imgNote4.setImageResource(0)
        imgNote5.setImageResource(0)
        imgNote6.setImageResource(0)
        imgNote7.setImageResource(0)
        selectedColor = "#ffd633"

        val intent = Intent("bottom_sheet_action")
        intent.putExtra("action","Yellow")
        intent.putExtra("selectedColor",selectedColor)

LocalBroadcastManager.getInstance(requireContext()).sendBroadcast(intent)

    }

    fNote4.setOnClickListener {
        imgNote1.setImageResource(0)
        imgNote2.setImageResource(0)
        imgNote4.setImageResource(R.drawable.ic_tick)
        imgNote5.setImageResource(0)
        imgNote6.setImageResource(0)
        imgNote7.setImageResource(0)
        selectedColor = "#ae3b76"

        val intent = Intent("bottom_sheet_action")
        intent.putExtra("action","Purple")
        intent.putExtra("selectedColor",selectedColor)

LocalBroadcastManager.getInstance(requireContext()).sendBroadcast(intent)

    }

    fNote5.setOnClickListener {
        imgNote1.setImageResource(0)
        imgNote2.setImageResource(0)
        imgNote4.setImageResource(0)
        imgNote5.setImageResource(R.drawable.ic_tick)
        imgNote6.setImageResource(0)
        imgNote7.setImageResource(0)
        selectedColor = "#0aebaf"

        val intent = Intent("bottom_sheet_action")
        intent.putExtra("action","Green")
        intent.putExtra("selectedColor",selectedColor)

LocalBroadcastManager.getInstance(requireContext()).sendBroadcast(intent)

    }
```

```
fNote6.setOnClickListener {  
  
    imgNote1.setImageResource(0)  
    imgNote2.setImageResource(0)  
    imgNote4.setImageResource(0)  
    imgNote5.setImageResource(0)  
    imgNote6.setImageResource(R.drawable.ic_tick)  
    imgNote7.setImageResource(0)  
    selectedColor = "#ff7746"  
    val intent = Intent("bottom_sheet_action")  
    intent.putExtra("action", "Orange")  
    intent.putExtra("selectedColor", selectedColor)  
  
LocalBroadcastManager.getInstance(requireContext()).sendBroadcast(intent)  
}  
  
fNote7.setOnClickListener {  
    imgNote1.setImageResource(0)  
    imgNote2.setImageResource(0)  
    imgNote4.setImageResource(0)  
    imgNote5.setImageResource(0)  
    imgNote6.setImageResource(0)  
    imgNote7.setImageResource(R.drawable.ic_tick)  
    selectedColor = "#202734"  
  
    val intent = Intent("bottom_sheet_action")  
    intent.putExtra("action", "Black")  
    intent.putExtra("selectedColor", selectedColor)  
  
LocalBroadcastManager.getInstance(requireContext()).sendBroadcast(intent)  
}  
  
layoutImage.setOnClickListener{  
    val intent = Intent("bottom_sheet_action")  
    intent.putExtra("action", "Image")  
  
LocalBroadcastManager.getInstance(requireContext()).sendBroadcast(intent)  
    dismiss()  
}  
  
layoutWebUrl.setOnClickListener{  
    val intent = Intent("bottom_sheet_action")  
    intent.putExtra("action", "WebUrl")  
  
LocalBroadcastManager.getInstance(requireContext()).sendBroadcast(intent)  
    dismiss()  
}  
  
layoutDeleteNote.setOnClickListener {  
    val intent = Intent("bottom_sheet_action")  
    intent.putExtra("action", "DeleteNote")  
  
LocalBroadcastManager.getInstance(requireContext()).sendBroadcast(intent)  
    dismiss()  
}  
}  
}
```

Basefragment.kt

```
import android.os.Bundle
import androidx.fragment.app.Fragment
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.Job
import kotlin.coroutines.CoroutineContext

abstract class BaseFragment : Fragment(), CoroutineScope {

    private lateinit var job: Job
    override val coroutineContext: CoroutineContext
        get() = job + Dispatchers.Main

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        job = Job()
    }

    override fun onDestroy() {
        super.onDestroy()
        job.cancel()
    }
}
```

CreateNoteFragment.kt

```
package com.codingwithme.notesapp

import android.Manifest
import android.app.Activity
import android.app.Activity.RESULT_OK
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.content.IntentFilter
import android.content.pm.PackageManager
import android.graphics.Bitmap
import android.graphics.BitmapFactory
import android.graphics.Color
import android.net.Uri
import android.os.Bundle
import android.os.PatternMatcher
import android.provider.MediaStore
import android.util.Patterns
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import androidx.fragment.app.Fragment
```

```
import androidx.localbroadcastmanager.content.LocalBroadcastManager
import com.codingwithme.notesapp.database.NotesDatabase
import com.codingwithme.notesapp.entities.Notes
import com.codingwithme.notesapp.util.NoteBottomSheetFragment
import com.theartofdev.edmodo.cropper.CropImage
import com.theartofdev.edmodo.cropper.CropImageView
import kotlinx.android.synthetic.main.fragment_create_note.*
import kotlinx.android.synthetic.main.fragment_home.*
import kotlinx.android.synthetic.main.item_rv_notes.view.*
import kotlinx.coroutines.launch
import pub.devrel.easypermissions.AppSettingsDialog
import pub.devrel.easypermissions.EasyPermissions
import java.text.SimpleDateFormat
import java.util.*
import java.util.regex.Pattern

class CreateNoteFragment :
    BaseFragment(), EasyPermissions.PermissionCallbacks, EasyPermissions.RationaleC
    allbacks {

    var selectedColor = "#171C26"
    var currentDate:String? = null
    private var READ_STORAGE_PERM = 123
    private var REQUEST_CODE_IMAGE = 456
    private var selectedImagePath = ""
    private var webLink = ""
    private var noteId = -1

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        noteId = requireArguments().getInt("noteId",-1)
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_create_note, container,
false)
    }

    companion object {
        @JvmStatic
        fun newInstance() =
            CreateNoteFragment().apply {
                arguments = Bundle().apply {
                }
            }
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
    }
}
```

```
        if (noteId != -1){

            launch {
                context?.let {
                    var notes =
NotesDatabase.getDatabase(it).noteDao().getSpecificNote(noteId)

colorView.setBackgroundColor(Color.parseColor(notes.color))
                    etNoteTitle.setText(notes.title)
                    etNoteSubTitle.setText(notes.subTitle)
                    etNoteDesc.setText(notes.noteText)
                    if (notes.imgPath != ""){
                        selectedImagePath = notes.imgPath!!
imgNote.setImageBitmap(BitmapFactory.decodeFile(notes.imgPath))
                        layoutImage.visibility = View.VISIBLE
                        imgNote.visibility = View.VISIBLE
                        imgDelete.visibility = View.VISIBLE
                    }else{
                        layoutImage.visibility = View.GONE
                        imgNote.visibility = View.GONE
                        imgDelete.visibility = View.GONE
                    }

                    if (notes.webLink != ""){
                        webLink = notes.webLink!!
                        tvWebLink.text = notes.webLink
                        layoutWebUrl.visibility = View.VISIBLE
                        etWebLink.setText(notes.webLink)
                        imgUrlDelete.visibility = View.VISIBLE
                    }else{
                        imgUrlDelete.visibility = View.GONE
                        layoutWebUrl.visibility = View.GONE
                    }
                }
            }
        }
        LocalBroadcastManager.getInstance(requireContext()).registerReceiver(
            BroadcastReceiver, IntentFilter("bottom_sheet_action")
        )

        val sdf = SimpleDateFormat("dd/M/yyyy hh:mm:ss")

        currentDate = sdf.format(Date())
        colorView.setBackgroundColor(Color.parseColor(selectedColor))

        tvDateTime.text = currentDate

        imgDone.setOnClickListener {
            if (noteId != -1){
                updateNote()
            }else{
                saveNote()
            }
        }
    }
}
```

```
imgBack.setOnClickListener {
    requireActivity().supportFragmentManager.popBackStack()
}

imgMore.setOnClickListener{

    var noteBottomSheetFragment =
NoteBottomSheetFragment.newInstance(noteId)

noteBottomSheetFragment.show(requireActivity().supportFragmentManager,"Note
Bottom Sheet Fragment")
}

imgDelete.setOnClickListener {
    selectedImagePath = ""
    layoutImage.visibility = View.GONE
}

btnOk.setOnClickListener {
    if (etWebLink.text.toString().trim().isEmpty()){
        checkWebUrl()
    }else{
        Toast.makeText(requireContext(),"Url is
Required",Toast.LENGTH_SHORT).show()
    }
}

btnCancel.setOnClickListener {
    if (noteId != -1){
        tvWebLink.visibility = View.VISIBLE
        layoutWebUrl.visibility = View.GONE
    }else{
        layoutWebUrl.visibility = View.GONE
    }
}

imgUrlDelete.setOnClickListener {
    webLink = ""
    tvWebLink.visibility = View.GONE
    imgUrlDelete.visibility = View.GONE
    layoutWebUrl.visibility = View.GONE
}

tvWebLink.setOnClickListener {
    var intent =
Intent(Intent.ACTION_VIEW,Uri.parse(etWebLink.text.toString()))
    startActivity(intent)
}

}
private fun updateNote() {
    launch {
```

```
        context?.let {
            var notes =
NotesDatabase.getDatabase(it).noteDao().getSpecificNote(noteId)

            notes.title = etNoteTitle.text.toString()
            notes.subTitle = etNoteSubTitle.text.toString()
            notes.noteText = etNoteDesc.text.toString()
            notes.dateTime = currentDate
            notes.color = selectedColor
            notes.imgPath = selectedImagePath
            notes.webLink = webLink

            NotesDatabase.getDatabase(it).noteDao().updateNote(notes)
            etNoteTitle.setText("")
            etNoteSubTitle.setText("")
            etNoteDesc.setText("")
            layoutImage.visibility = View.GONE
            imgNote.visibility = View.GONE
            tvWebLink.visibility = View.GONE
            requireActivity().supportFragmentManager.popBackStack()
        }
    }
}

private fun saveNote(){

    if (etNoteTitle.text.isNullOrEmpty()){
        Toast.makeText(context,"Note Title is
Required",Toast.LENGTH_SHORT).show()
    }
    else if (etNoteSubTitle.text.isNullOrEmpty()){

        Toast.makeText(context,"Note Sub Title is
Required",Toast.LENGTH_SHORT).show()
    }

    else if (etNoteDesc.text.isNullOrEmpty()){

        Toast.makeText(context,"Note Description is
Required",Toast.LENGTH_SHORT).show()
    }

    else{

        launch {
            var notes = Notes()
            notes.title = etNoteTitle.text.toString()
            notes.subTitle = etNoteSubTitle.text.toString()
            notes.noteText = etNoteDesc.text.toString()
            notes.dateTime = currentDate
            notes.color = selectedColor
            notes.imgPath = selectedImagePath
            notes.webLink = webLink
            context?.let {
                NotesDatabase.getDatabase(it).noteDao().insertNotes(notes)
                etNoteTitle.setText("")
                etNoteSubTitle.setText("")
                etNoteDesc.setText("")
            }
        }
    }
}
```

```
        layoutImage.visibility = View.GONE
        imgNote.visibility = View.GONE
        tvWebLink.visibility = View.GONE
        requireActivity().supportFragmentManager.popBackStack()
    }
}
}
}
private fun deleteNote() {
    launch {
        context?.let {
            NotesDatabase.getDatabase(it).noteDao().deleteSpecificNote(noteId)
            requireActivity().supportFragmentManager.popBackStack()
        }
    }
}
private fun checkWebUrl() {
    if (Patterns.WEB_URL.matcher(etWebLink.text.toString()).matches()) {
        layoutWebUrl.visibility = View.GONE
        etWebLink.isEnabled = false
        webLink = etWebLink.text.toString()
        tvWebLink.visibility = View.VISIBLE
        tvWebLink.text = etWebLink.text.toString()
    } else {
        Toast.makeText(requireContext(), "Url is not
valid", Toast.LENGTH_SHORT).show()
    }
}
private val BroadcastReceiver : BroadcastReceiver = object
:BroadcastReceiver() {
    override fun onReceive(p0: Context?, p1: Intent?) {
        var actionColor = p1!!.getStringExtra("action")
        when(actionColor!!) {
            "Blue" -> {
                selectedColor = p1!!.getStringExtra("selectedColor")!!
                colorView.setBackgroundColor(Color.parseColor(selectedColor))
            }
            "Yellow" -> {
                selectedColor = p1!!.getStringExtra("selectedColor")!!
                colorView.setBackgroundColor(Color.parseColor(selectedColor))
            }
            "Purple" -> {
                selectedColor = p1!!.getStringExtra("selectedColor")!!
                colorView.setBackgroundColor(Color.parseColor(selectedColor))
            }
        }
    }
}
```



```
        }
        "Green" -> {
            selectedColor = p1.getStringExtra("selectedColor")!!
colorView.setBackgroundColor(Color.parseColor(selectedColor))

        }

        "Orange" -> {
            selectedColor = p1.getStringExtra("selectedColor")!!
colorView.setBackgroundColor(Color.parseColor(selectedColor))

        }

        "Black" -> {
            selectedColor = p1.getStringExtra("selectedColor")!!
colorView.setBackgroundColor(Color.parseColor(selectedColor))

        }

        "Image" ->{
            readStorageTask()
            layoutWebUrl.visibility = View.GONE
        }

        "WebUrl" ->{
            layoutWebUrl.visibility = View.VISIBLE
        }
        "DeleteNote" -> {
            //delete note
            deleteNote()
        }

        else -> {
            layoutImage.visibility = View.GONE
            imgNote.visibility = View.GONE
            layoutWebUrl.visibility = View.GONE
            selectedColor = p1.getStringExtra("selectedColor")!!
colorView.setBackgroundColor(Color.parseColor(selectedColor))

        }
    }
}

override fun onDestroy() {

LocalBroadcastManager.getInstance(requireContext()).unregisterReceiver(BroadcastReceiver)
    super.onDestroy()
}
```

```
    }

    private fun hasReadStoragePerm(): Boolean {
        return
EasyPermissions.hasPermissions(requireContext(), Manifest.permission.READ_EXTERNAL_STORAGE)
    }

    private fun readStorageTask() {
        if (hasReadStoragePerm()) {
            pickImageFromGallery()
        } else {
            EasyPermissions.requestPermissions(
                requireActivity(),
                getString(R.string.storage_permission_text),
                READ_STORAGE_PERM,
                Manifest.permission.READ_EXTERNAL_STORAGE
            )
        }
    }

    private fun pickImageFromGallery() {
        var intent =
Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI)
        if (intent.resolveActivity(requireActivity().packageManager) !=
null) {
            startActivityForResult(intent, REQUEST_CODE_IMAGE)
        }
    }

    private fun getPathFromUri(contentUri: Uri): String? {
        var filePath: String? = null
        var cursor =
requireActivity().contentResolver.query(contentUri, null, null, null, null)
        if (cursor == null) {
            filePath = contentUri.path
        } else {
            cursor.moveToFirst()
            var index = cursor.getColumnIndex("_data")
            filePath = cursor.getString(index)
            cursor.close()
        }
        return filePath
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data:
Intent?) {
        super.onActivityResult(requestCode, resultCode, data)
        if (requestCode == REQUEST_CODE_IMAGE && resultCode == RESULT_OK) {
            if (data != null) {
                var selectedImageUrl = data.data
                if (selectedImageUrl != null) {
                    try {
                        var inputStream =
requireActivity().contentResolver.openInputStream(selectedImageUrl)
                    }
                }
            }
        }
    }
}
```

```
        var bitmap = BitmapFactory.decodeStream(inputStream)
        imgNote.setImageBitmap(bitmap)
        imgNote.visibility = View.VISIBLE
        layoutImage.visibility = View.VISIBLE

        selectedImagePath =
getPathFromUri(selectedImageUrl)!!
    } catch (e: Exception) {

Toast.makeText(requireContext(), e.message, Toast.LENGTH_SHORT).show()
    }

    }

    }

    }

    override fun onRequestPermissionsResult(
        requestCode: Int,
        permissions: Array<out String>,
        grantResults: IntArray
    ) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults)

EasyPermissions.onRequestPermissionsResult(requestCode, permissions, grantResul
ts, requireActivity())
    }

    override fun onPermissionsDenied(requestCode: Int, perms:
MutableList<String>) {
        if
(EasyPermissions.somePermissionPermanentlyDenied(requireActivity(), perms)) {
            AppSettingsDialog.Builder(requireActivity()).build().show()
        }
    }

    override fun onPermissionsGranted(requestCode: Int, perms:
MutableList<String>) {

    }

    override fun onRationaleDenied(requestCode: Int) {

    }

    override fun onRationaleAccepted(requestCode: Int) {

    }

}
```

HomeFragment.kt

```
import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.SearchView
import android.widget.Toast
import androidx.recyclerview.widget.StaggeredGridLayoutManager
import com.codingwithme.notesapp.adapter.NotesAdapter
import com.codingwithme.notesapp.database.NotesDatabase
import com.codingwithme.notesapp.entities.Notes
import kotlinx.android.synthetic.main.fragment_create_note.*
import kotlinx.android.synthetic.main.fragment_home.*
import kotlinx.coroutines.launch
import java.util.*
import kotlin.collections.ArrayList

class HomeFragment : BaseFragment() {

    var arrNotes = ArrayList<Notes>()
    var notesAdapter: NotesAdapter = NotesAdapter()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        arguments?.let {
        }
    }
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_home, container, false)
    }
    companion object {
        @JvmStatic
        fun newInstance() =
            HomeFragment().apply {
                arguments = Bundle().apply {
                }
            }
    }
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        recycler_view.setHasFixedSize(true)

        recycler_view.layoutManager =
            StaggeredGridLayoutManager(2, StaggeredGridLayoutManager.VERTICAL)

        launch {
            context?.let {
                var notes =
                    NotesDatabase.getDatabase(it).noteDao().getAllNotes()
            }
        }
    }
}
```

```

        notesAdapter!!.setData(notes)
        arrNotes = notes as ArrayList<Notes>
        recycler_view.adapter = notesAdapter
    }
}
notesAdapter!!.setOnClickListener(onClicked)

fabBtnCreateNote.setOnClickListener {
    replaceFragment(CreateNoteFragment.newInstance(), false)
}
search_view.setOnQueryTextListener( object :
SearchView.OnQueryTextListener{
    override fun onQueryTextSubmit(p0: String?): Boolean {
        return true
    }
    override fun onQueryTextChange(p0: String?): Boolean {

        var tempArr = ArrayList<Notes>()

        for (arr in arrNotes){
            if
(arr.title!!.toLowerCase(Locale.getDefault()).contains(p0.toString())){
                tempArr.add(arr)
            }
        }
        notesAdapter.setData(tempArr)
        notesAdapter.notifyDataSetChanged()
        return true
    }
})
}
private val onClicked = object :NotesAdapter.OnItemClickListener{
    override fun onClicked(notesId: Int) {
        var fragment :Fragment
        var bundle = Bundle()
        bundle.putInt("noteId",notesId)
        fragment = CreateNoteFragment.newInstance()
        fragment.arguments = bundle

        replaceFragment(fragment, false)
    }
}
fun replaceFragment(fragment:Fragment, istransition:Boolean){
    val fragmentTransition =
activity!!.supportFragmentManager.beginTransaction()

    if (istransition){
fragmentTransition.setCustomAnimations(android.R.anim.slide_out_right,android
.R.anim.slide_in_left)
    }

    fragmentTransition.replace(R.id.frame_layout,fragment).addToBackStack(fragment
t.javaClass.simpleName).commit()
}
}

```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:background="@color/colorPrimaryDark"
    tools:context=".MainActivity">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:id="@+id/frame_layout"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

fragment_create_note.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:background="@color/colorPrimaryDark"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imgBack"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/_10sdp"
        android:layout_marginTop="@dimen/_20sdp"
        android:layout_marginBottom="@dimen/_10sdp"
        android:src="@drawable/ic_back"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:tint="@color/ColorWhite" />

    <ImageView
        android:id="@+id/imgDone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/_20sdp"
```

```
        android:layout_marginEnd="@dimen/_10sdp"
        android:layout_marginBottom="@dimen/_10sdp"
        android:src="@drawable/ic_tick"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:tint="@color/ColorWhite" />

<EditText
    android:id="@+id/etNoteTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/_10sdp"
    android:background="@null"
    android:ellipsize="end"
    android:fontFamily="@font/manrope_bold"
    android:hint="@string/notes_title"
    android:inputType="text"
    android:maxLines="1"
    android:padding="@dimen/_10sdp"
    android:textColor="@color/ColorWhite"
    android:textColorHint="@color/ColorGray"
    android:textSize="@dimen/_14ssp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/imgBack" />

<TextView
    android:id="@+id/tvDateTime"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="@dimen/_10sdp"
    android:layout_marginTop="@dimen/_5sdp"
    android:layout_marginEnd="@dimen/_10sdp"
    android:layout_marginBottom="@dimen/_5sdp"
    android:ellipsize="end"
    android:fontFamily="@font/manrope_bold"
    android:maxLines="1"
    android:paddingStart="@dimen/_10sdp"
    android:textColor="@color/ColorGray"
    android:textColorHint="@color/ColorGray"
    android:textSize="@dimen/_8ssp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/etNoteTitle" />

<LinearLayout
    android:id="@+id/ll"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/_10sdp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/tvDateTime">
```

```
<View
    android:id="@+id/colorView"
    android:layout_width="@dimen/_5sdp"
    android:layout_height="match_parent"
    android:layout_marginStart="@dimen/_5sdp"
    android:background="@color/ColorBlue" />

<EditText
    android:id="@+id/etNoteSubTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@null"
    android:ellipsize="end"
    android:fontFamily="@font/manrope_regular"
    android:hint="@string/notes_sub_title"
    android:inputType="text"
    android:maxLines="1"
    android:padding="@dimen/_10sdp"
    android:textColor="@color/ColorWhite"
    android:textColorHint="@color/ColorGray"
    android:textSize="@dimen/_10ssp" />
</LinearLayout>

<RelativeLayout
    android:id="@+id/layoutImage"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:visibility="gone"
    app:layout_constraintTop_toBottomOf="@id/l1">

    <ImageView
        android:id="@+id/imgNote"
        android:layout_width="match_parent"
        android:layout_height="@dimen/_300sdp"
        android:visibility="gone"
        android:layout_marginStart="@dimen/_10sdp"
        android:layout_marginTop="@dimen/_10sdp"
        android:layout_marginEnd="@dimen/_10sdp" />

    <ImageView
        android:id="@+id/imgDelete"
        android:layout_width="@dimen/_24sdp"
        android:layout_height="@dimen/_24sdp"
        android:layout_marginEnd="@dimen/_15sdp"
        android:layout_marginTop="@dimen/_20sdp"
        android:layout_alignParentEnd="true"
        android:visibility="gone"
        android:layout_alignParentTop="true"
        android:src="@drawable/ic_delete" />

</RelativeLayout>

<LinearLayout
```



```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/layoutWebUrl"
        android:background="@color/colorPrimary"
        app:layout_constraintEnd_toEndOf="parent"
        android:padding="@dimen/_10sdp"
        android:visibility="gone"
        android:layout_margin="@dimen/_10sdp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/layoutImage"
        android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:orientation="horizontal">
        <EditText
            android:id="@+id/etWebLink"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:background="@null"
            android:fontFamily="@font/manrope_regular"
            android:hint="@string/web_url"
            android:inputType="textMultiLine"
            android:padding="@dimen/_10sdp"
            android:text="http://"
            android:textColor="@color/ColorWhite"
            android:textColorHint="@color/ColorGray"
            android:textSize="@dimen/_12ssp" />
        <ImageView
            android:id="@+id/imgUrlDelete"
            android:layout_width="@dimen/_24sdp"
            android:layout_height="@dimen/_24sdp"
            android:visibility="gone"
            android:src="@drawable/ic_baseline_cancel_24" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="end"
        android:layout_marginTop="@dimen/_10sdp"
        android:gravity="end"
        android:orientation="horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="@dimen/_30sdp"
            android:fontFamily="@font/manrope_bold"
            android:background="@color/colorAccent"
            android:text="@string/cancel"
            android:textSize="@dimen/_12ssp"
```

```
        android:textAllCaps="false"
        android:textColor="@color/ColorWhite"
        android:id="@+id/btnCancel"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="@dimen/_30sdp"
            android:fontFamily="@font/manrope_bold"
            android:background="@color/colorAccent"
            android:text="@string/ok"
            android:textAllCaps="false"
            android:textSize="@dimen/_12ssp"
            android:layout_marginStart="@dimen/_5sdp"
            android:textColor="@color/ColorWhite"
            android:id="@+id/btnOk"/>

    </LinearLayout>
</LinearLayout>

<TextView
    android:id="@+id/tvWebLink"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="@dimen/_10sdp"
    android:layout_marginTop="@dimen/_5sdp"
    android:layout_marginEnd="@dimen/_10sdp"
    android:layout_marginBottom="@dimen/_5sdp"
    android:ellipsize="end"
    android:visibility="gone"
    android:fontFamily="@font/manrope_bold"
    android:padding="@dimen/_10sdp"
    android:textColor="@color/ColorGray"
    android:textColorHint="@color/ColorGray"
    android:textSize="@dimen/_12ssp"
    android:autoLink="web"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutWebUrl" />

<EditText
    android:id="@+id/etNoteDesc"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/_10sdp"
    android:background="@null"
    android:fontFamily="@font/manrope_regular"
    android:hint="@string/notes_desc"
    android:inputType="textMultiLine"
    android:padding="@dimen/_10sdp"
    android:textColor="@color/ColorWhite"
    android:textColorHint="@color/ColorGray"
    android:textSize="@dimen/_12ssp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/tvWebLink" />
```

```
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="@dimen/_40sdp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent">

    <ImageView
        android:id="@+id/imgMore"
        android:layout_width="@dimen/_24sdp"
        android:layout_height="@dimen/_24sdp"
        android:layout_gravity="center"
        android:src="@drawable/img_bar" />

</FrameLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

fragment_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:background="@color/colorPrimaryDark"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imgBack"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/_10sdp"
        android:layout_marginTop="@dimen/_20sdp"
        android:layout_marginBottom="@dimen/_10sdp"
        android:src="@drawable/ic_back"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:tint="@color/ColorWhite" />

    <ImageView
        android:id="@+id/imgDone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/_20sdp"
        android:layout_marginEnd="@dimen/_10sdp"
        android:layout_marginBottom="@dimen/_10sdp"
        android:src="@drawable/ic_tick"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:tint="@color/ColorWhite" />
```

```
<EditText
    android:id="@+id/etNoteTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/_10sdp"
    android:background="@null"
    android:ellipsize="end"
    android:fontFamily="@font/manrope_bold"
    android:hint="@string/notes_title"
    android:inputType="text"
    android:maxLines="1"
    android:padding="@dimen/_10sdp"
    android:textColor="@color/ColorWhite"
    android:textColorHint="@color/ColorGray"
    android:textSize="@dimen/_14ssp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/imgBack" />

<TextView
    android:id="@+id/tvDateTime"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="@dimen/_10sdp"
    android:layout_marginTop="@dimen/_5sdp"
    android:layout_marginEnd="@dimen/_10sdp"
    android:layout_marginBottom="@dimen/_5sdp"
    android:ellipsize="end"
    android:fontFamily="@font/manrope_bold"
    android:maxLines="1"
    android:paddingStart="@dimen/_10sdp"
    android:textColor="@color/ColorGray"
    android:textColorHint="@color/ColorGray"
    android:textSize="@dimen/_8ssp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/etNoteTitle" />

<LinearLayout
    android:id="@+id/l1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/_10sdp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/tvDateTime">

    <View
        android:id="@+id/colorView"
        android:layout_width="@dimen/_5sdp"
        android:layout_height="match_parent"
        android:layout_marginStart="@dimen/_5sdp"
        android:background="@color/ColorBlue" />
```

```
<EditText
    android:id="@+id/etNoteSubTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@null"
    android:ellipsize="end"
    android:fontFamily="@font/manrope_regular"
    android:hint="@string/notes_sub_title"
    android:inputType="text"
    android:maxLines="1"
    android:padding="@dimen/_10sdp"
    android:textColor="@color/ColorWhite"
    android:textColorHint="@color/ColorGray"
    android:textSize="@dimen/_10ssp" />
</LinearLayout>

<RelativeLayout
    android:id="@+id/layoutImage"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:visibility="gone"
    app:layout_constraintTop_toBottomOf="@id/l1">

    <ImageView
        android:id="@+id/imgNote"
        android:layout_width="match_parent"
        android:layout_height="@dimen/_300sdp"
        android:visibility="gone"
        android:layout_marginStart="@dimen/_10sdp"
        android:layout_marginTop="@dimen/_10sdp"
        android:layout_marginEnd="@dimen/_10sdp" />

    <ImageView
        android:id="@+id/imgDelete"
        android:layout_width="@dimen/_24sdp"
        android:layout_height="@dimen/_24sdp"
        android:layout_marginEnd="@dimen/_15sdp"
        android:layout_marginTop="@dimen/_20sdp"
        android:layout_alignParentEnd="true"
        android:visibility="gone"
        android:layout_alignParentTop="true"
        android:src="@drawable/ic_delete" />

</RelativeLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/layoutWebUrl"
    android:background="@color/colorPrimary"
    app:layout_constraintEnd_toEndOf="parent"
    android:padding="@dimen/_10sdp"
    android:visibility="gone"
```

```
        android:layout_margin="@dimen/_10sdp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/layoutImage"
        android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:orientation="horizontal">
        <EditText
            android:id="@+id/etWebLink"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:background="@null"
            android:fontFamily="@font/manrope_regular"
            android:hint="@string/web_url"
            android:inputType="textMultiLine"
            android:padding="@dimen/_10sdp"
            android:text="http://"
            android:textColor="@color/ColorWhite"
            android:textColorHint="@color/ColorGray"
            android:textSize="@dimen/_12ssp" />
        <ImageView
            android:id="@+id/imgUrlDelete"
            android:layout_width="@dimen/_24sdp"
            android:layout_height="@dimen/_24sdp"
            android:visibility="gone"
            android:src="@drawable/ic_baseline_cancel_24" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="end"
        android:layout_marginTop="@dimen/_10sdp"
        android:gravity="end"
        android:orientation="horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="@dimen/_30sdp"
            android:fontFamily="@font/manrope_bold"
            android:background="@color/colorAccent"
            android:text="@string/cancel"
            android:textSize="@dimen/_12ssp"
            android:textAllCaps="false"
            android:textColor="@color/ColorWhite"
            android:id="@+id/btnCancel"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="@dimen/_30sdp"
```

```
        android:fontFamily="@font/manrope_bold"
        android:background="@color/colorAccent"
        android:text="@string/ok"
        android:textAllCaps="false"
        android:textSize="@dimen/_12ssp"
        android:layout_marginStart="@dimen/_5sdp"
        android:textColor="@color/ColorWhite"
        android:id="@+id/btnOk"/>

    </LinearLayout>
</LinearLayout>

<TextView
    android:id="@+id/tvWebLink"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="@dimen/_10sdp"
    android:layout_marginTop="@dimen/_5sdp"
    android:layout_marginEnd="@dimen/_10sdp"
    android:layout_marginBottom="@dimen/_5sdp"
    android:ellipsize="end"
    android:visibility="gone"
    android:fontFamily="@font/manrope_bold"
    android:padding="@dimen/_10sdp"
    android:textColor="@color/ColorGray"
    android:textColorHint="@color/ColorGray"
    android:textSize="@dimen/_12ssp"
    android:autoLink="web"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/layoutWebUrl" />

<EditText
    android:id="@+id/etNoteDesc"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/_10sdp"
    android:background="@null"
    android:fontFamily="@font/manrope_regular"
    android:hint="@string/notes_desc"
    android:inputType="textMultiLine"
    android:padding="@dimen/_10sdp"
    android:textColor="@color/ColorWhite"
    android:textColorHint="@color/ColorGray"
    android:textSize="@dimen/_12ssp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/tvWebLink" />

<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="@dimen/_40sdp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent">
```

```

        <ImageView
            android:id="@+id/imgMore"
            android:layout_width="@dimen/_24sdp"
            android:layout_height="@dimen/_24sdp"
            android:layout_gravity="center"
            android:src="@drawable/img_bar" />
    </FrameLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

fragment_note_button_sheet.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/layoutBottomSheet"
    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="wrap_content"
    android:orientation="vertical"
    app:behavior_peekHeight="40dp"
    app:layout_behavior="@string/bottom_sheet_behavior"
    android:background="@color/colorPrimary"
    tools:context=".MainActivity">
    <FrameLayout
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        android:layout_width="match_parent"
        android:layout_height="@dimen/_40sdp">

        <ImageView
            android:id="@+id/imgMore"
            android:layout_gravity="center"
            android:layout_width="@dimen/_24sdp"
            android:layout_height="@dimen/_24sdp"
            android:src="@drawable/img_bar"/>
    </FrameLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="@dimen/_5sdp"
        android:gravity="center"
        android:orientation="horizontal">

        <FrameLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:paddingStart="@dimen/_5sdp"
            android:id="@+id/fNote1">

            <View

```



```
        android:layout_width="@dimen/_24sdp"
        android:layout_height="@dimen/_24sdp"
        android:background="@drawable/note_color_1"/>

        <ImageView
            android:id="@+id/imgNote1"
            android:layout_width="@dimen/_16sdp"
            android:layout_height="@dimen/_16sdp"
            android:layout_gravity="center"
            app:tint="@color/ColorWhite"/>
    </FrameLayout>

    <FrameLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingStart="@dimen/_5sdp"
        android:id="@+id/fNote2">

        <View
            android:layout_width="@dimen/_24sdp"
            android:layout_height="@dimen/_24sdp"
            android:background="@drawable/note_color_2"/>

        <ImageView
            android:id="@+id/imgNote2"
            android:layout_width="@dimen/_16sdp"
            android:layout_height="@dimen/_16sdp"
            android:layout_gravity="center"
            app:tint="@color/ColorWhite"/>
    </FrameLayout>

    <FrameLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingStart="@dimen/_5sdp"
        android:id="@+id/fNote4">

        <View
            android:layout_width="@dimen/_24sdp"
            android:layout_height="@dimen/_24sdp"
            android:background="@drawable/note_color_4"/>

        <ImageView
            android:id="@+id/imgNote4"
            android:layout_width="@dimen/_16sdp"
            android:layout_height="@dimen/_16sdp"
            android:layout_gravity="center"
            app:tint="@color/ColorWhite"/>
    </FrameLayout>

    <FrameLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingStart="@dimen/_5sdp"
        android:id="@+id/fNote5">
```

```
<View
    android:layout_width="@dimen/_24sdp"
    android:layout_height="@dimen/_24sdp"
    android:background="@drawable/note_color_5"/>

    <ImageView
        android:id="@+id/imgNote5"
        android:layout_width="@dimen/_16sdp"
        android:layout_height="@dimen/_16sdp"
        android:layout_gravity="center"
        app:tint="@color/ColorWhite"/>
</FrameLayout>

<FrameLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingStart="@dimen/_5sdp"
    android:id="@+id/fNote6">

    <View
        android:layout_width="@dimen/_24sdp"
        android:layout_height="@dimen/_24sdp"
        android:background="@drawable/note_color_6"/>

    <ImageView
        android:id="@+id/imgNote6"
        android:layout_width="@dimen/_16sdp"
        android:layout_height="@dimen/_16sdp"
        android:layout_gravity="center"
        app:tint="@color/ColorWhite"/>
</FrameLayout>
<FrameLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingStart="@dimen/_5sdp"
    android:id="@+id/fNote7">

    <View
        android:layout_width="@dimen/_24sdp"
        android:layout_height="@dimen/_24sdp"
        android:background="@drawable/note_color_7"/>

    <ImageView
        android:id="@+id/imgNote7"
        android:layout_width="@dimen/_16sdp"
        android:layout_height="@dimen/_16sdp"
        android:layout_gravity="center"
        app:tint="@color/ColorWhite"/>
</FrameLayout>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAlignment="textEnd"
    android:paddingEnd="@dimen/_10sdp"
    app:layout_constraintTop_toBottomOf="@id/etNoteTitle"
    app:layout_constraintStart_toStartOf="parent"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        android:textColor="@color/ColorWhite"
        android:textSize="@dimen/_12ssp"
        android:text="Pick Color"
        android:fontFamily="@font/manrope_bold"
        android:maxLines="1"
        android:ellipsize="end"
    />
</LinearLayout>

<LinearLayout
    android:id="@+id/layoutImage"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:background="@color/colorPrimary"
    android:gravity="center"
    android:layout_margin="@dimen/_10sdp"
    android:orientation="horizontal">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_image"
        android:layout_marginStart="@dimen/_5sdp"
        android:layout_marginEnd="@dimen/_10sdp"
        app:tint="@color/ColorWhite" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@id/etNoteTitle"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:textColor="@color/ColorWhite"
        android:textSize="@dimen/_12ssp"
        android:text="Add Image"
        android:fontFamily="@font/manrope_bold"
        android:maxLines="1"
        android:ellipsize="end"
    />

</LinearLayout>
<LinearLayout
    android:id="@+id/layoutWebUrl"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:background="@color/colorPrimary"
    android:gravity="center"
    android:layout_marginStart="@dimen/_10sdp"
    android:layout_marginEnd="@dimen/_10sdp"
```

```
        android:layout_marginBottom="@dimen/_10sdp"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/ic_link"
            android:layout_marginStart="@dimen/_5sdp"
            android:layout_marginEnd="@dimen/_10sdp"
            app:tint="@color/ColorWhite" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:layout_constraintTop_toBottomOf="@id/etNoteTitle"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            android:textColor="@color/ColorWhite"
            android:textSize="@dimen/_12ssp"
            android:text="Add Web Link"
            android:fontFamily="@font/manrope_bold"
            android:maxLines="1"
            android:ellipsize="end"
            />

    </LinearLayout>
    <LinearLayout
        android:id="@+id/layoutDeleteNote"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:background="@color/colorPrimary"
        android:gravity="center"
        android:visibility="gone"
        android:layout_marginStart="@dimen/_10sdp"
        android:layout_marginEnd="@dimen/_10sdp"
        android:layout_marginBottom="@dimen/_10sdp"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/ic_delete"
            android:layout_marginStart="@dimen/_5sdp"
            android:layout_marginEnd="@dimen/_10sdp"
            app:tint="@color/ColorWhite" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:layout_constraintTop_toBottomOf="@id/etNoteTitle"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
```

```
        android:textColor="@color/ColorWhite"
        android:textSize="@dimen/_12ssp"
        android:text="Delete Note"
        android:fontFamily="@font/manrope_bold"
        android:maxLines="1"
        android:ellipsize="end"
    />
</LinearLayout>
</LinearLayout>
```

item_rv_notes.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:id="@+id/layoutNote"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_height="wrap_content">

    <androidx.cardview.widget.CardView
        android:id="@+id/cardView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardCornerRadius="@dimen/_10sdp"
        app:cardElevation="@dimen/_5sdp"
        android:layout_margin="@dimen/_5sdp"
        app:cardBackgroundColor="@color/ColorLightBlack"
        android:padding="@dimen/_5sdp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <com.makeramen.roundedimageview.RoundedImageView
                android:id="@+id/imgNote"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:adjustViewBounds="true"
                android:visibility="gone"
                app:riv_corner_radius_top_left="@dimen/_10sdp"
                app:riv_corner_radius_top_right="@dimen/_10sdp"/>

            <TextView
                android:id="@+id/tvTitle"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textSize="@dimen/_15ssp"
                android:text="@string/notes"
                android:paddingStart="@dimen/_10sdp"
                android:paddingTop="@dimen/_10sdp"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintStart_toStartOf="parent"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        android:fontFamily="@font/manrope_bold"
        android:textColor="@color/ColorWhite"/>

    <TextView
        android:id="@+id/tvWebLink"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/_5sdp"
        android:layout_marginBottom="@dimen/_5sdp"
        android:ellipsize="end"
        android:visibility="gone"
        android:fontFamily="@font/manrope_bold"
        android:padding="@dimen/_10sdp"
        android:textColor="@color/ColorGray"
        android:textColorHint="@color/ColorGray"
        android:textSize="@dimen/_12ssp"
        android:autoLink="web" />

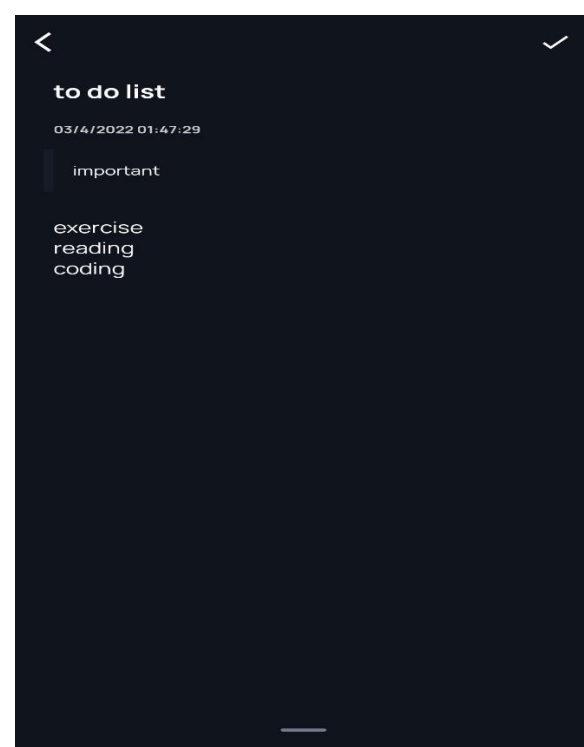
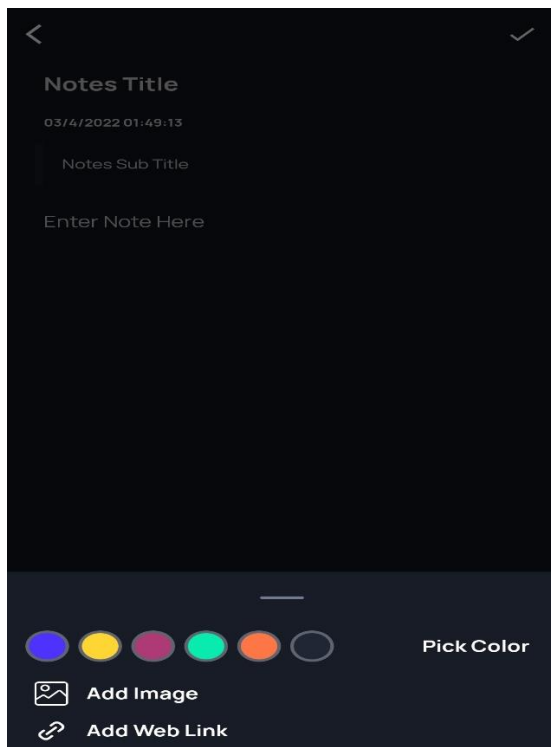
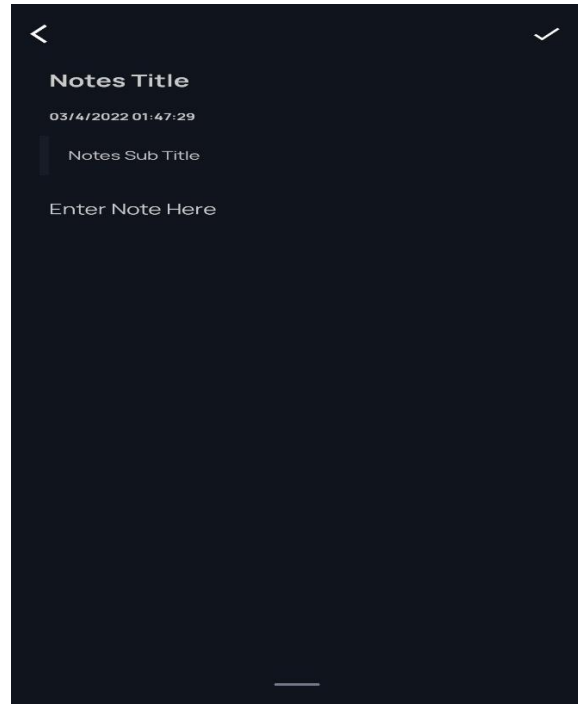
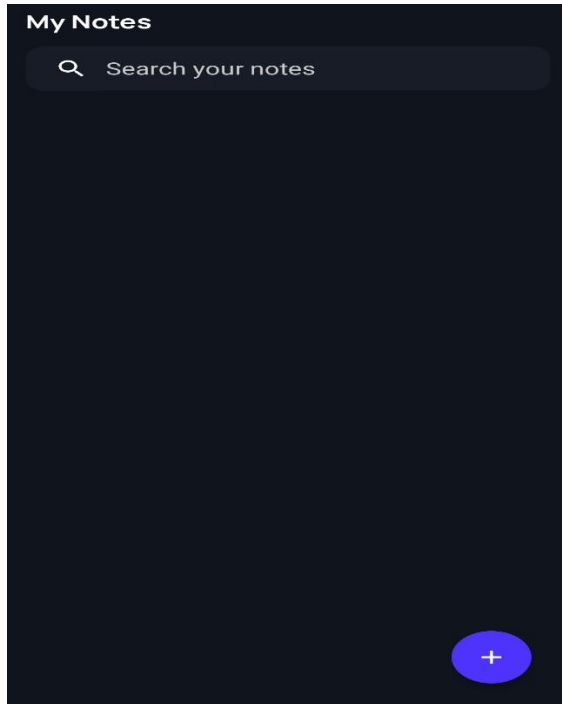
    <TextView
        android:id="@+id/tvDesc"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="@dimen/_11ssp"
        android:text="@string/notes"
        android:paddingStart="@dimen/_10sdp"
        android:paddingTop="@dimen/_5sdp"
        android:paddingBottom="@dimen/_5sdp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:fontFamily="@font/manrope_regular"
        android:textColor="@color/ColorText"/>

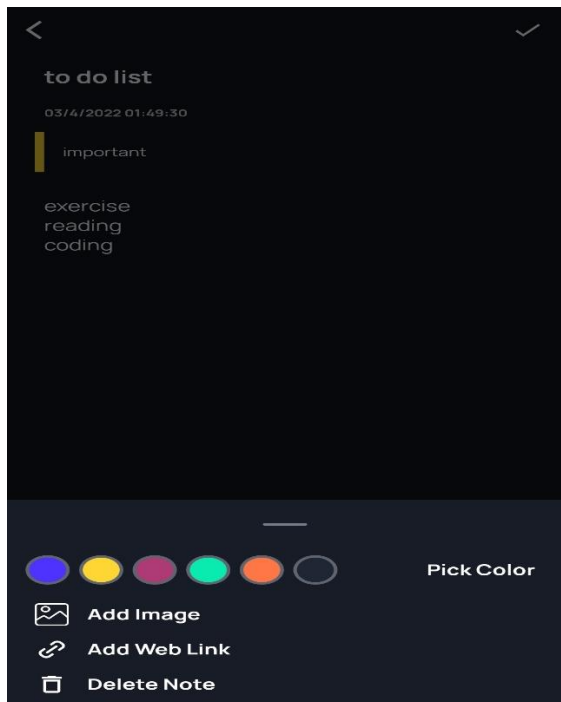
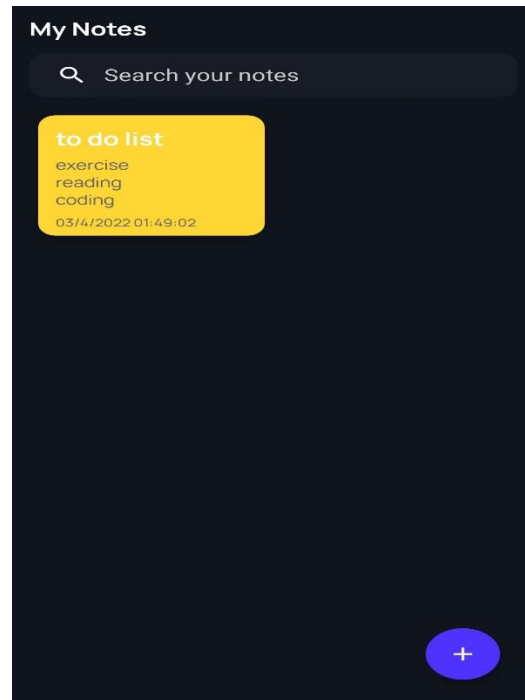
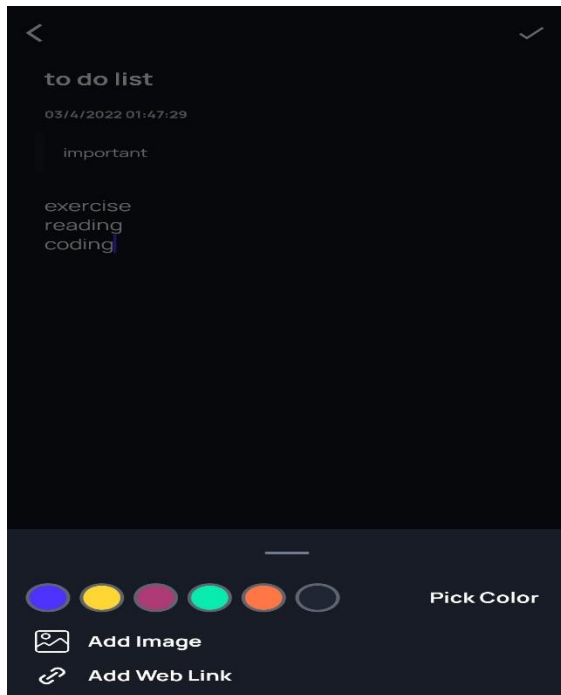
    <TextView
        android:id="@+id/tvDateTime"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="@dimen/_9ssp"
        android:text="@string/notes"
        android:paddingStart="@dimen/_10sdp"
        android:paddingTop="@dimen/_2sdp"
        android:paddingBottom="@dimen/_5sdp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:fontFamily="@font/manrope_regular"
        android:textColor="@color/ColorText"/>

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

</LinearLayout>
```

Results





REFERENCE

<https://www.w3schools.in/android/tutorials/>

<https://www.geeksforgeeks.org/android-studio-tutorial/>

<https://www.youtube.com/watch?v=dx7TWA9Oot4>