**TITLE: A MATERIAL DESIGN STUDY APP**

**TEAM MEMBERS:**

**SANTHOSH S - 812022104067**

**SRIRAM C - 812022104076**

**SUSTHIK RAJA - 812022104082**

**MADHAN KUMAR-812022104306**

**Description:**

A **Material Design study app** is an educational tool designed to help users organize, manage, and track their learning activities through an intuitive and visually appealing interface inspired by Google's Material Design principles. The app aims to provide a streamlined, user-friendly experience that makes studying more efficient and enjoyable by offering easy navigation, interactive elements, and personalized learning tools.  
The core features of this app would include:

* **Organized Study Materials**:
* Users can categorize study materials by subject, topic, or lesson, with visually distinct cards and lists for easy browsing and access.
* **Task Management**:
* The app allows users to set tasks, reminders, and study goals, helping them stay on track with their study plans.
* **Study Timers**:
* A built-in timer or Pomodoro clock helps users manage study sessions and break times, encouraging focused, productive learning.
* **Progress Tracking**:
* Visual progress bars or charts track the user's learning milestones, showing how much study material has been completed, and how much is left.
* **Flashcards and Quizzes**:
* Interactive flashcards and quizzes test the user's knowledge, offering immediate feedback to reinforce learning.
* **Material Design Aesthetics**:
* The app uses Material Design’s principles of depth, color, and motion to create an engaging and tactile experience. Clean typography, smooth animations, and a consistent color palette guide the user through the app’s features seamlessly.

**MainActivity.Kt :**

package com.example.owlapplication

import android.content.Context

import android.content.Intent

import android.os.Bundle

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.Image

import androidx.compose.foundation.clickable

import androidx.compose.foundation.layout.\*

import androidx.compose.foundation.rememberScrollState

import androidx.compose.foundation.verticalScroll

import androidx.compose.material.Card

import androidx.compose.material.Text

import androidx.compose.runtime.Composable

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.draw.scale

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.res.painterResource

import androidx.compose.ui.res.stringResource

import androidx.compose.ui.text.font.FontWeight

import androidx.compose.ui.text.style.TextAlign

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

StudyApp(this)

}

}

}

@Composable

fun StudyApp(context: Context) {

Column(

modifier = Modifier

.padding(20.dp)

.verticalScroll(rememberScrollState())

) {

Text(text = "Study Material",

fontSize = 36.sp,

fontWeight = FontWeight.Bold,

color = Color(0xFFFFA500),

modifier = Modifier.align(Alignment.CenterHorizontally))

Spacer(modifier = Modifier.height(20.dp))

// 01

Card(

modifier = Modifier

.fillMaxWidth()

.height(250.dp)

.clickable {

context.startActivity(

Intent(context, MainActivity2::class.java)

)

},

elevation = 8.dp

)

{

Column(

horizontalAlignment = Alignment.CenterHorizontally

) {

Image(

painterResource(id = R.drawable.img\_1), contentDescription = "",

modifier = Modifier

.height(150.dp)

.scale(scaleX = 1.2F, scaleY = 1F)

)

Text(text = stringResource(id = R.string.course1),color = Color(0xFFFFA500),

fontSize = 16.sp)

Text(

text = stringResource(id = R.string.topic1),

fontWeight = FontWeight.Bold,

fontSize = 20.sp,

textAlign = TextAlign.Center,

)

}

}

Spacer(modifier = Modifier.height(20.dp))

// 02

Card(

modifier = Modifier

.fillMaxWidth()

.height(250.dp)

.clickable {

context.startActivity(

Intent(context, MainActivity3::class.java)

)

},

elevation = 8.dp

)

{

Column(

horizontalAlignment = Alignment.CenterHorizontally

) {

Image(

painterResource(id = R.drawable.img\_2), contentDescription = "",

modifier = Modifier

.height(150.dp)

.scale(scaleX = 1.4F, scaleY = 1F)

)

Text(text = stringResource(id = R.string.course2),color = Color(0xFFFFA500),

fontSize = 16.sp)

Text(

text = stringResource(id = R.string.topic2),

fontWeight = FontWeight.Bold,

fontSize = 20.sp,

textAlign = TextAlign.Center,

)

}

}

Spacer(modifier = Modifier.height(20.dp))

// 03

Card(

modifier = Modifier

.fillMaxWidth()

.height(250.dp)

.clickable {

context.startActivity(

Intent(context, MainActivity4::class.java)

)

},

elevation = 8.dp

)

{

Column(

horizontalAlignment = Alignment.CenterHorizontally

) {

Image(

painterResource(id = R.drawable.img\_3), contentDescription = "",

modifier = Modifier

.height(150.dp)

.scale(scaleX = 1.2F, scaleY = 1F)

)

Text(text = stringResource(id = R.string.course3),color = Color(0xFFFFA500),

fontSize = 16.sp)

Text(

text = stringResource(id = R.string.topic3),

fontWeight = FontWeight.Bold,

fontSize = 20.sp,

textAlign = TextAlign.Center,

)

}

}

Spacer(modifier = Modifier.height(20.dp))

// 04

Card(

modifier = Modifier

.fillMaxWidth()

.height(250.dp)

.clickable {

context.startActivity(

Intent(context, MainActivity5::class.java)

)

},

elevation = 8.dp

)

{

Column(

horizontalAlignment = Alignment.CenterHorizontally

) {

Image(

painterResource(id = R.drawable.img\_4), contentDescription = "",

modifier = Modifier

.height(150.dp)

.scale(scaleX = 1.2F, scaleY = 1F)

)

Text(text = stringResource(id = R.string.course4),color = Color(0xFFFFA500),

fontSize = 16.sp)

Text(

text = stringResource(id = R.string.topic4),

fontWeight = FontWeight.Bold,

fontSize = 20.sp,

textAlign = TextAlign.Center,

)

}

}

}

}

**Mainactivity2.Kt:**

package com.example.owlapplication

import android.os.Bundle

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.Image

import androidx.compose.foundation.background

import androidx.compose.foundation.layout.\*

import androidx.compose.foundation.rememberScrollState

import androidx.compose.foundation.verticalScroll

import androidx.compose.material.Text

import androidx.compose.runtime.Composable

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.draw.scale

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.res.painterResource

import androidx.compose.ui.res.stringResource

import androidx.compose.ui.text.font.FontWeight

import androidx.compose.ui.text.style.TextAlign

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import com.example.owlapplication.ui.theme.OwlApplicationTheme

class MainActivity2 : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

Greeting()

}

}

}

@Composable

fun Greeting() {

Column(

modifier = Modifier.padding(start = 26.dp, end = 26.dp, bottom = 26.dp)

.verticalScroll(rememberScrollState())

.background(Color.White),

verticalArrangement = Arrangement.Top

) {

Image(

painterResource(id = R.drawable.img\_1),

contentDescription = "",

modifier = Modifier.align(Alignment.CenterHorizontally)

.scale(scaleX = 1.5F, scaleY = 1.5F)

)

Spacer(modifier = Modifier.height(60.dp))

Text(

text = stringResource(id = R.string.course1),

color = Color(0xFFFFA500),

fontSize = 16.sp,

modifier = Modifier.align(Alignment.CenterHorizontally)

)

Spacer(modifier = Modifier.height(20.dp))

Text(

text = stringResource(id = R.string.topic1),

fontWeight = FontWeight.Bold,

fontSize = 26.sp,

modifier = Modifier.align(Alignment.CenterHorizontally)

)

Spacer(modifier = Modifier.height(20.dp))

Text(

text = stringResource(id = R.string.subheading1\_1),

modifier = Modifier.align(Alignment.Start),

fontSize = 20.sp

)

Spacer(modifier = Modifier.height(20.dp))

Text(

text = stringResource(id = R.string.text1\_1),

modifier = Modifier.align(Alignment.Start),

textAlign = TextAlign.Justify,

fontSize = 16.sp

)

Spacer(modifier = Modifier.height(20.dp))

Text(

text = stringResource(id = R.string.subheading1\_2),

modifier = Modifier.align(Alignment.Start),

fontSize = 20.sp

)

Spacer(modifier = Modifier.height(20.dp))

Text(

text = stringResource(id = R.string.text1\_2),

modifier = Modifier.align(Alignment.Start),

textAlign = TextAlign.Justify,

fontSize = 16.sp

)

}

}

**UserDatabaseHelper.Kt:**

package com.example.owlapplication

import android.annotation.SuppressLint

import android.content.ContentValues

import android.content.Context

import android.database.Cursor

import android.database.sqlite.SQLiteDatabase

import android.database.sqlite.SQLiteOpenHelper

class UserDatabaseHelper(context: Context) :

SQLiteOpenHelper(context, DATABASE\_NAME, null, DATABASE\_VERSION) {

companion object {

private const val DATABASE\_VERSION = 1

private const val DATABASE\_NAME = "UserDatabase.db"

private const val TABLE\_NAME = "user\_table"

private const val COLUMN\_ID = "id"

private const val COLUMN\_FIRST\_NAME = "first\_name"

private const val COLUMN\_LAST\_NAME = "last\_name"

private const val COLUMN\_EMAIL = "email"

private const val COLUMN\_PASSWORD = "password"

}

override fun onCreate(db: SQLiteDatabase?) {

val createTable = "CREATE TABLE $TABLE\_NAME (" +

"$COLUMN\_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +

"$COLUMN\_FIRST\_NAME TEXT, " +

"$COLUMN\_LAST\_NAME TEXT, " +

"$COLUMN\_EMAIL TEXT, " +

"$COLUMN\_PASSWORD TEXT" +

")"

db?.execSQL(createTable)

}

override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {

db?.execSQL("DROP TABLE IF EXISTS $TABLE\_NAME")

onCreate(db)

}

fun insertUser(user: User) {

val db = writableDatabase

val values = ContentValues()

values.put(COLUMN\_FIRST\_NAME, user.firstName)

values.put(COLUMN\_LAST\_NAME, user.lastName)

values.put(COLUMN\_EMAIL, user.email)

values.put(COLUMN\_PASSWORD, user.password)

db.insert(TABLE\_NAME, null, values)

db.close()

}

@SuppressLint("Range")

fun getUserByUsername(username: String): User? {

val db = readableDatabase

val cursor: Cursor = db.rawQuery("SELECT \* FROM $TABLE\_NAME WHERE $COLUMN\_FIRST\_NAME = ?", arrayOf(username))

var user: User? = null

if (cursor.moveToFirst()) {

user = User(

id = cursor.getInt(cursor.getColumnIndex(COLUMN\_ID)),

firstName = cursor.getString(cursor.getColumnIndex(COLUMN\_FIRST\_NAME)),

lastName = cursor.getString(cursor.getColumnIndex(COLUMN\_LAST\_NAME)),

email = cursor.getString(cursor.getColumnIndex(COLUMN\_EMAIL)),

password = cursor.getString(cursor.getColumnIndex(COLUMN\_PASSWORD)),

)

}

cursor.close()

db.close()

return user

}

@SuppressLint("Range")

fun getUserById(id: Int): User? {

val db = readableDatabase

val cursor: Cursor = db.rawQuery("SELECT \* FROM $TABLE\_NAME WHERE $COLUMN\_ID = ?", arrayOf(id.toString()))

var user: User? = null

if (cursor.moveToFirst()) {

user = User(

id = cursor.getInt(cursor.getColumnIndex(COLUMN\_ID)),

firstName = cursor.getString(cursor.getColumnIndex(COLUMN\_FIRST\_NAME)),

lastName = cursor.getString(cursor.getColumnIndex(COLUMN\_LAST\_NAME)),

email = cursor.getString(cursor.getColumnIndex(COLUMN\_EMAIL)),

password = cursor.getString(cursor.getColumnIndex(COLUMN\_PASSWORD)),

)

}

cursor.close()

db.close()

return user

}

@SuppressLint("Range")

fun getAllUsers(): List<User> {

val users = mutableListOf<User>()

val db = readableDatabase

val cursor: Cursor = db.rawQuery("SELECT \* FROM $TABLE\_NAME", null)

if (cursor.moveToFirst()) {

do {

val user = User(

id = cursor.getInt(cursor.getColumnIndex(COLUMN\_ID)),

firstName = cursor.getString(cursor.getColumnIndex(COLUMN\_FIRST\_NAME)),

lastName = cursor.getString(cursor.getColumnIndex(COLUMN\_LAST\_NAME)),

email = cursor.getString(cursor.getColumnIndex(COLUMN\_EMAIL)),

password = cursor.getString(cursor.getColumnIndex(COLUMN\_PASSWORD)),

)

users.add(user)

} while (cursor.moveToNext())

}

cursor.close()

db.close()

return users

}

}

**Output:**





