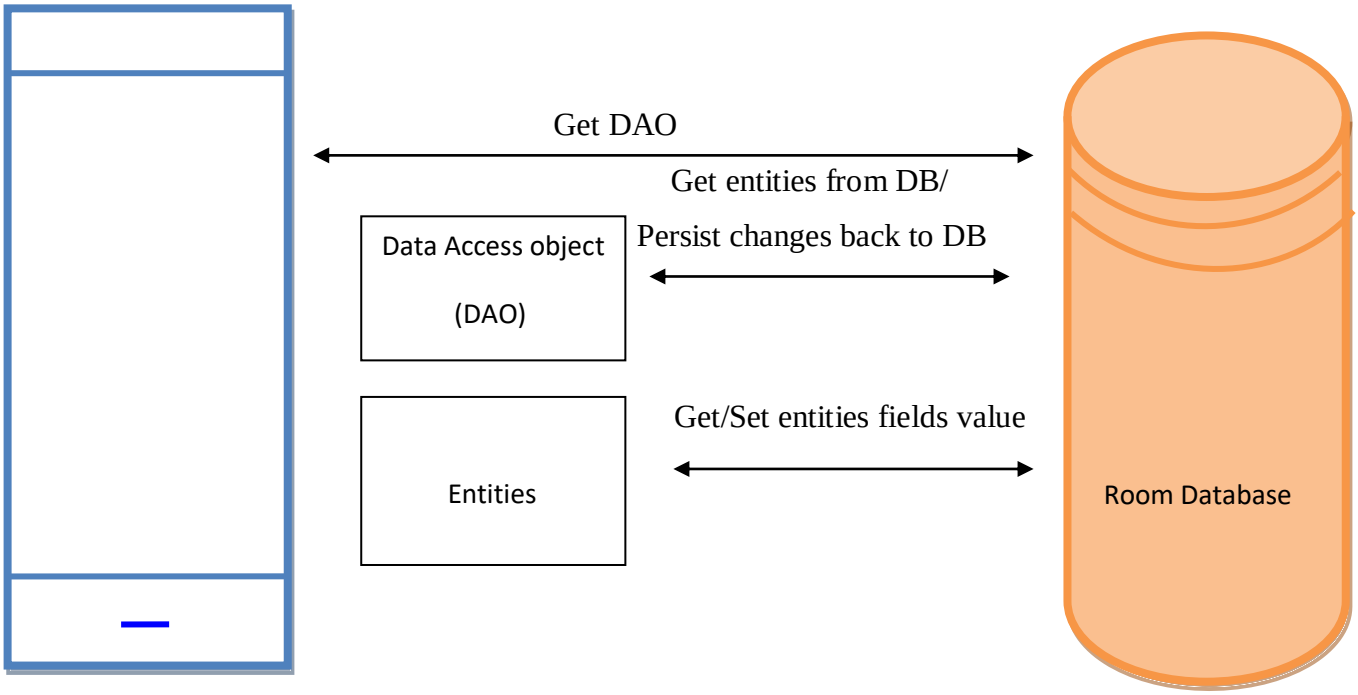
INTRODUCTION

1.1 OVERVIEW

A project that demonstrates the use of Android Jetpack Compose to build a UI for a Owl-M, a material design study app. Owl-M app is a sample project built using the Android Compose UI toolkit. A Compose implementation of the Owl Material study.

Architecture



Android app

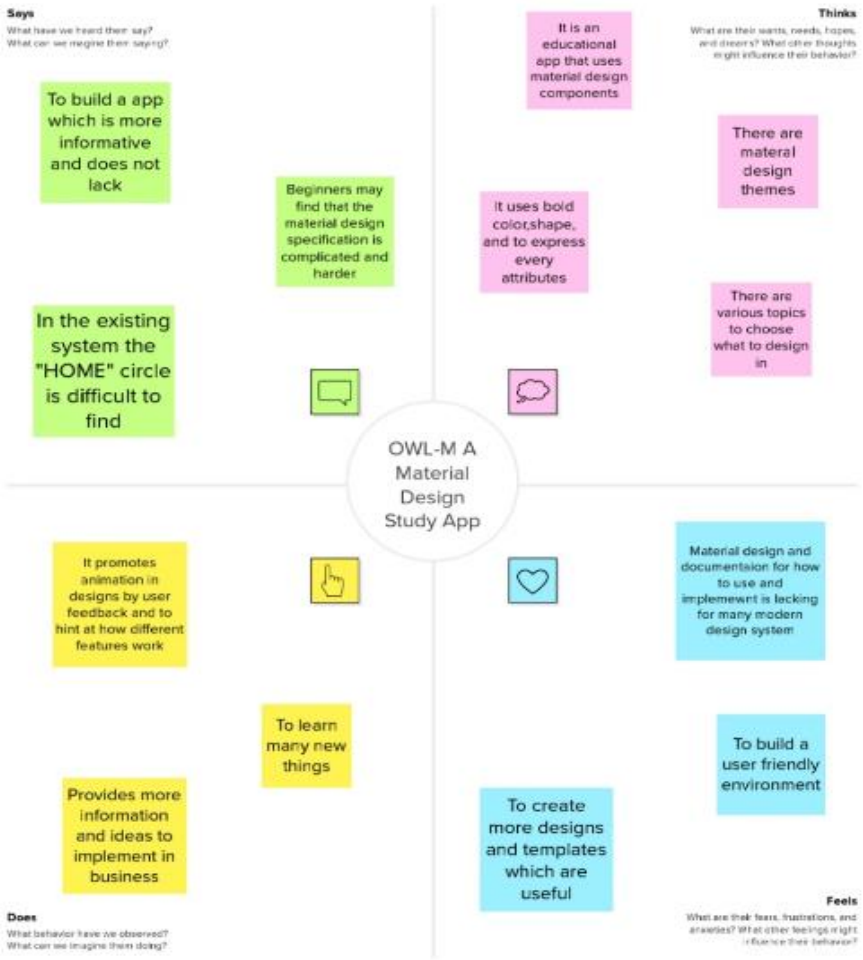
1.1 PURPOSE

Owl is an educational app that provides courses for people who want to explore and learn new skills in design, art, architecture, and fashion. The Owl uses bold color, shape and typography to express its attributes. Enable designers to quickly build apps that are usable. By doing this project we are able to work on android studio and build an app.

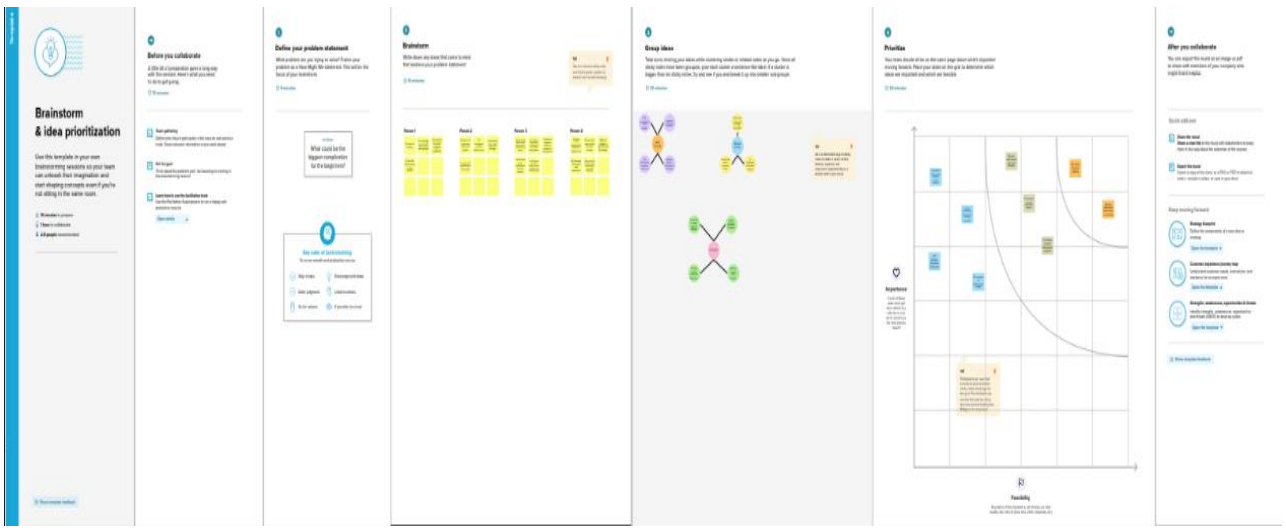
- We can able to integrate the database accordingly.
- User can view all the subject themes on selecting the themes he can read about it.
- It provides information about various topics.

1. PROBLEM DEFINITION & DESIGN THINKING

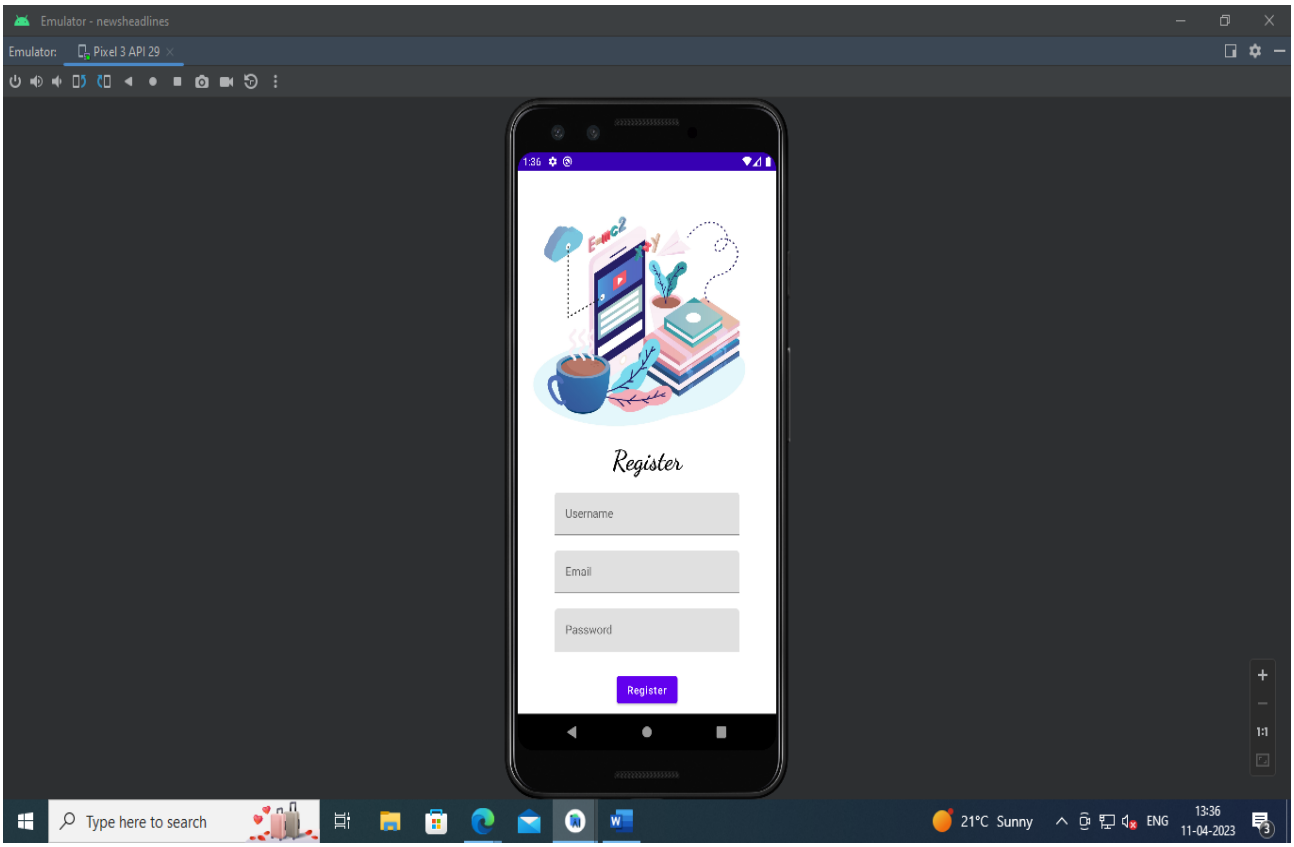
2.1Empathy Map

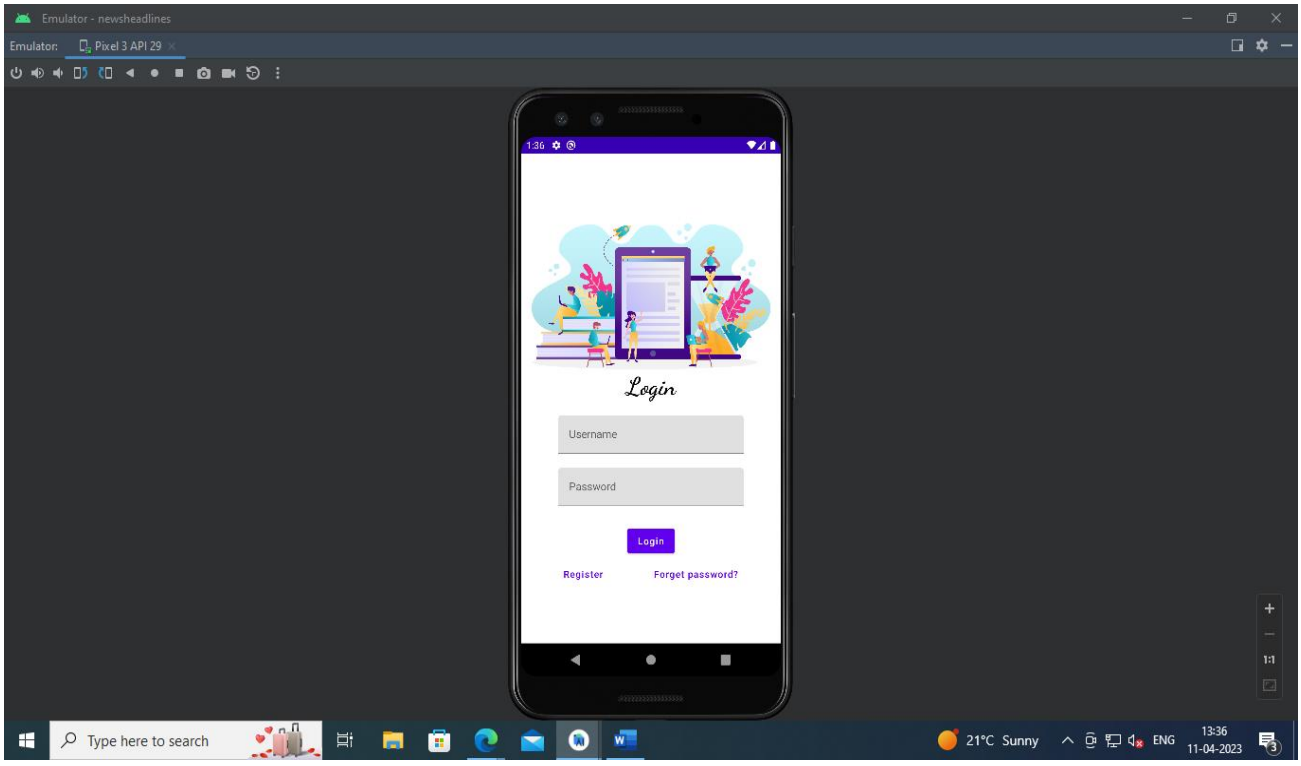


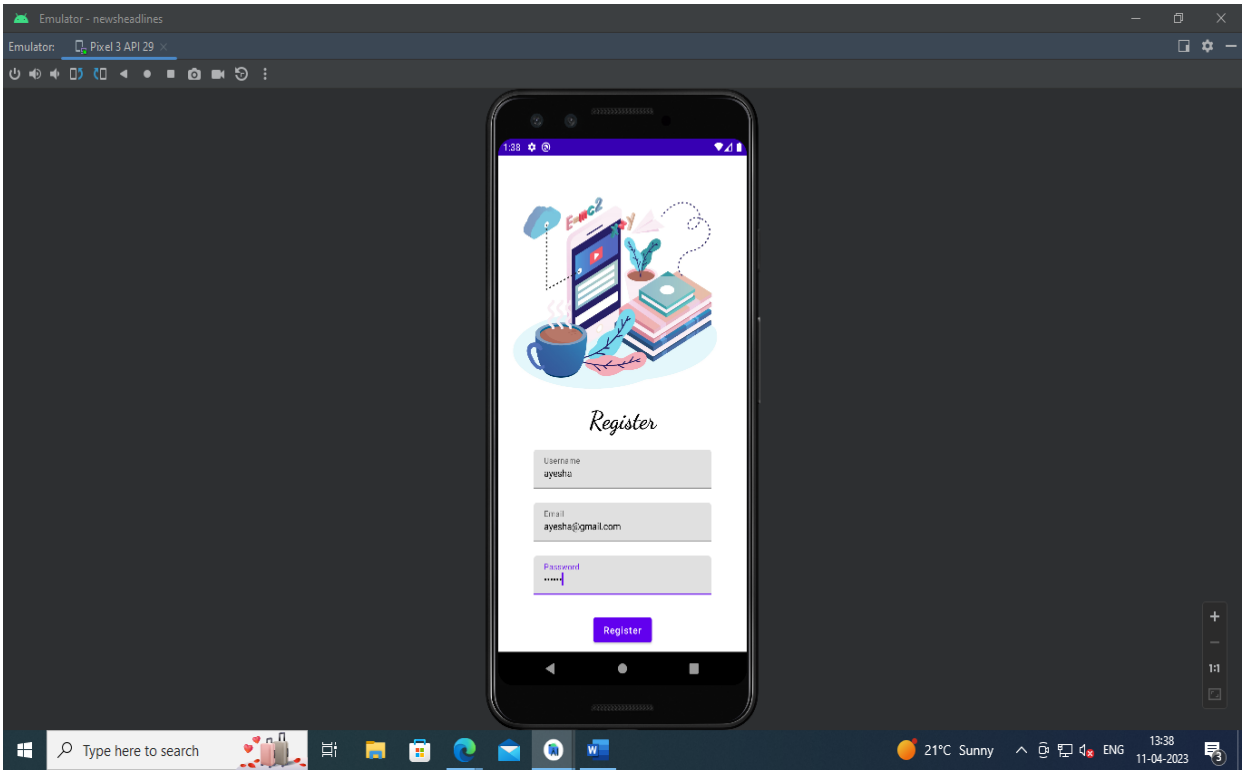
2.2Ideation & Brainstroming map

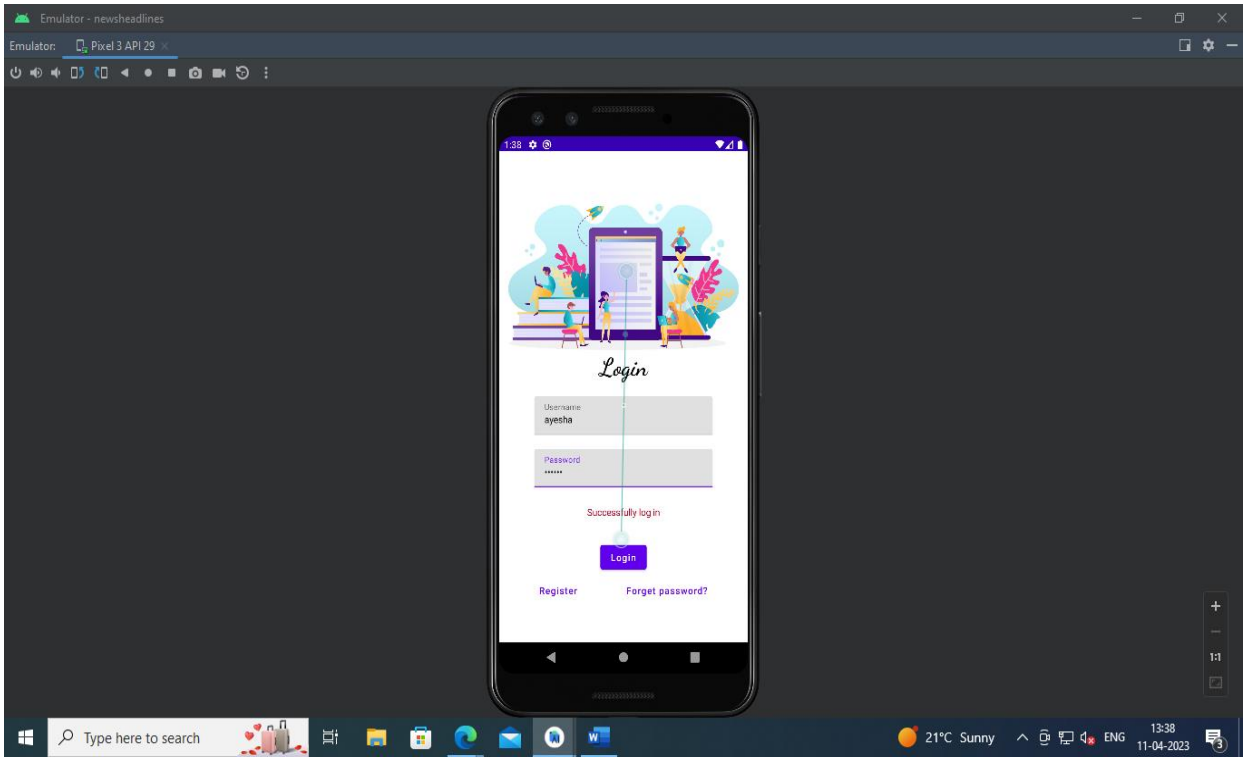


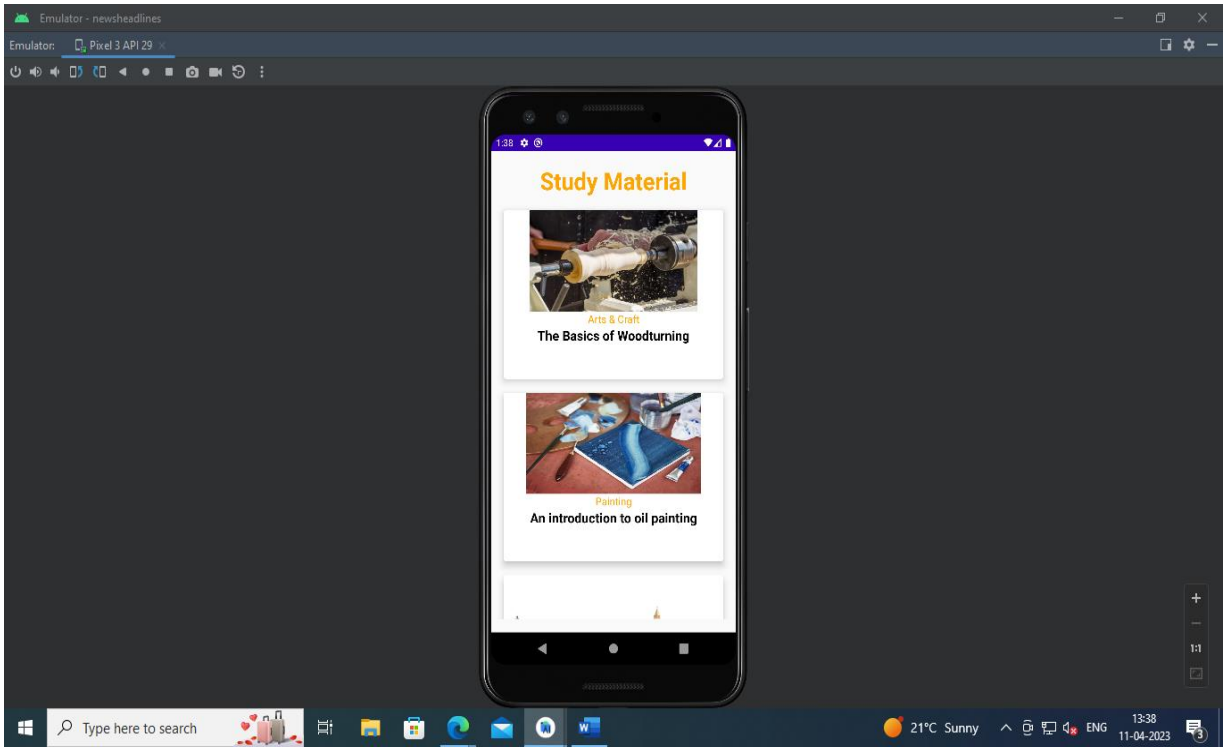
3. RESULT

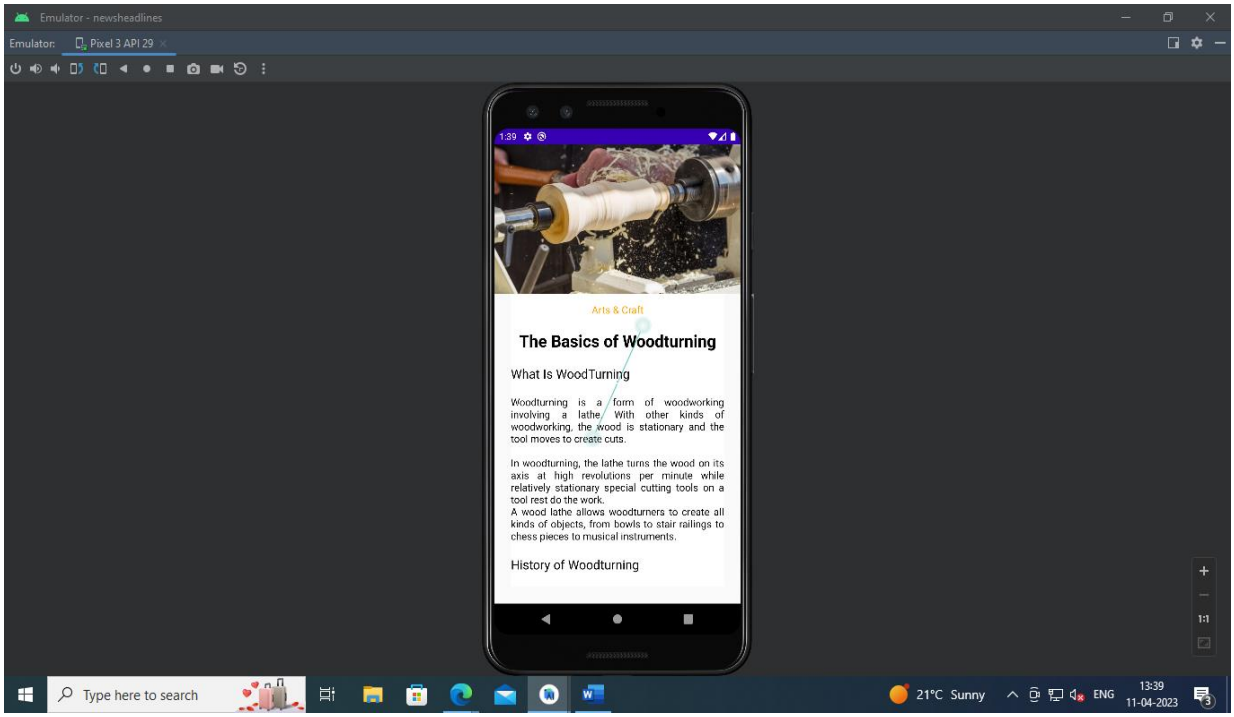

Emulator - newsheadlines
Emulator: Pixel 3 API 29
1:36
Register
Username
Email
Password
Register
Type here to search
21°C Sunny
ENG
13:36
11-04-2023


Emulator - newsheadlines
Emulator: Pixel 3 API 29
1:36
Login
Username
Password
Login
Register
Forget password?
Type here to search
21°C Sunny
ENG
13:36
11-04-2023


Emulator - newsheadlines
Emulator: Pixel 3 API 29
1:38
Register
Username
ayesha
Email
ayesha@gmail.com
Password
Register
Type here to search
21°C Sunny
ENG
13:38
11-04-2023


Emulator - newsheadlines
Emulator: Pixel 3 API 29
1:38
Login
Username
ayesha
Password
Successfully login
Login
Register
Forget password?
Type here to search
21°C Sunny
ENG
13:38
11-04-2023


Emulator - newsheadlines
Emulator: Pixel 3 API 29
1:38
Study Material
Arts & Craft
The Basics of Woodturning
Painting
An introduction to oil painting
Type here to search
21°C Sunny
ENG
13:38
11-04-2023


Emulator - newsheadlines
Emulator: Pixel 3 API 29
1:39
Arts & Craft
The Basics of Woodturning
What Is WoodTurning
Woodturning is a form of woodworking involving a lathe. With other kinds of woodworking, the wood is stationary and the tool moves to create cuts.
In woodturning, the lathe turns the wood on its axis at high revolutions per minute while relatively stationary special cutting tools on a tool rest do the work.
A wood lathe allows woodturners to create all kinds of objects, from bowls to stair railings to chess pieces to musical instruments.
History of Woodturning
Type here to search
21°C Sunny
ENG
13:39
11-04-2023

4. ADVANTAGES

- Material design come with graceful animations making it easier for users to understand the relationship between different elements on their devices.
- The use of fast page-switching animations helps users navigate through different screens without having to wait for the screen to load.
- Material design introduces self-contained UI components that are much easier to maintain and recognize.
- Material design theming which helps developers get started by choosing from a set of pre-designed UI components and customizing them according to their needs.
- Material Design takes care of the responsiveness part, providing your app with a uniform look, so users can navigate through their screens easily
- Less time required to create animations
- Developers can easily customize this new design language by using color, typography, and graphic elements that best suit their needs. It is flexible.

DISADVANTAGES

- It becomes difficult for users to look at screens that contain too much color.
- The objects move to different positions on the screen when a user performs an action this can be distracting for users.
- There are too many distractions for users due to that they can't focus on one thing.
- It is a learning curve, users may get confusions and difficult.

5. APPLICATIONS

Since it is an educational app it can be used in:

- Schools
- Colleges
- Designing industries
- Geological field
- Art and Architecture

6. CONCLUSION

If an app is being built primarily for the Android platform, then using Material Design is an easy choice. This app based on Material Design principles is going to feel like a native app. There are plenty of other use cases outside of the Android platform where Material Design is a solid choice. As the design system matures even further, those situations are bound to increase. Designers should, at the very least, familiarize themselves with the guidelines so that they can determine for themselves and it's appropriate to use Material Design. Material Design is a comprehensive set of design guidelines that goes beyond a simple framework. This study material android application ensures that students are able to access they required study materials which are essential.

7. FUTURE SCOPE

The application can be enhanced by improving the chat assistant so that it can clarify most of the doubts of the designers. The functionality to check the availability of a particular book of the university can be added to the application also all the study materials can be displayed. It is referred to the process of management which coordinates, supervises and executes the tasks associated with the flow of materials. Future this application may integrate different AI features such as voice recognition.

8. APPENDIX

SOURCE CODE

GRADLE SCRIPT

```
plugins {
    id 'com.android.application'
id 'org.jetbrains.kotlin.android'
}

android {
    namespace 'com.example.owlapplication'
compileSdk 33

defaultConfig {
        applicationId "com.example.owlapplication"
minSdk 22
targetSdk 33
versionCode 1
versionName "1.0"

testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
vectorDrawables {
            useSupportLibrary true
}
    }

    buildTypes {
        release {
            minifyEnabled false
proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
'proguard-rules.pro'
}
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = '1.8'
}
    buildFeatures {
        compose true
}
    composeOptions {
        kotlinCompilerExtensionVersion '1.2.0'
}
    packagingOptions {
        resources {
            excludes += '/META-INF/{AL2.0,LGPL2.1}'
}
    }
}

dependencies {

    implementation 'androidx.core:core-ktx:1.7.0'
```

```
implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.3.1'
implementation 'androidx.activity:activity-compose:1.3.1'
implementation "androidx.compose.ui:ui:$compose_ui_version"
implementation "androidx.compose.ui:ui-tooling-preview:$compose_ui_version"
implementation 'androidx.compose.material:material:1.2.0'
testImplementation 'junit:junit:4.13.2'
androidTestImplementation 'androidx.test.ext:junit:1.1.5'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
androidTestImplementation "androidx.compose.ui:ui-test-
junit4:$compose_ui_version"
debugImplementation "androidx.compose.ui:ui-tooling:$compose_ui_version"
debugImplementation "androidx.compose.ui:ui-test-
manifest:$compose_ui_version"
implementation 'androidx.room:room-common:2.5.0'

implementation 'androidx.room:room-ktx:2.5.0'

}
```

CREATING DATABASE CLASSES

USER DATA CLASS

```
package com.example.owlapplication
import androidx.room.ColumnInfo
import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity(tableName = "user_table")
data class User( @PrimaryKey(autoGenerate = true) val id: Int?,
@ColumnInfo(name = "first_name") val firstName: String?,
@ColumnInfo(name = "last_name") val lastName: String?,
@ColumnInfo(name = "email") val email: String?,
@ColumnInfo(name = "password") val password: String?,)
```

USER DAO INTERFACE

```
package com.example.owlapplication
import androidx.room.*

@Dao
interface UserDao {
    @Query("SELECT * FROM user_table WHERE email = :email")
suspend fun getUserByEmail(email: String): User?
    @Insert(onConflict = OnConflictStrategy.REPLACE)
suspend fun insertUser(user: User)
    @Update
suspend fun updateUser(user: User)

    @Delete
suspend fun deleteUser(user: User)
}
```

USER DATABASE CLASS

```
package com.example.owlapplication
import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase

@Database(entities = [User::class], version = 1)
abstract class UserDatabase : RoomDatabase() {

abstract fun userDao(): UserDao

companion object {

@Volatile
private var instance: UserDatabase? = null

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

USER DATABASE HELPER CLASS

```
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
    }

}
```

BUILDING APPLICATION UI AND CONNECTING TO DATABASE

LOGIN ACTIVITY KT WITH DATABASE

```
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.background
import androidx.compose.foundation.layout.*
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.layout.ContentScale
```

```
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat
//import com.example.owlapplication.ui.theme.OwlApplicationTheme

class LoginActivity : ComponentActivity() {
private lateinit var databaseHelper: UserDatabaseHelper
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(this)
        setContent {
LoginScreen(this, databaseHelper)
}
}
}
@Composable
fun LoginScreen(context: Context, databaseHelper: UserDatabaseHelper) {

var username by remember { mutableStateOf("") }
var password by remember { mutableStateOf("") }
var error by remember { mutableStateOf("") }

Column(
modifier = Modifier.fillMaxSize().background(Color.White),
horizontalAlignment = Alignment.CenterHorizontally,
verticalArrangement = Arrangement.Center
    ) {

Image(painterResource(id = R.drawable.study_login), contentDescription =
"")

        Text(
fontSize = 36.sp,
fontWeight = FontWeight.ExtraBold,
fontFamily = FontFamily.Cursive,
text = "Login"
)
        Spacer(modifier = Modifier.height(10.dp))

        TextField(
value = username,
onValueChange = { username = it },
label = { Text("Username") },
modifier = Modifier.padding(10.dp)
                .width(280.dp)
        )

        TextField(
value = password,
onValueChange = { password = it },
label = { Text("Password") },
visualTransformation = PasswordVisualTransformation(),
modifier = Modifier.padding(10.dp)
                .width(280.dp)
        )
```

```
if (error.isNotEmpty()) {
            Text(
text = error,
color = MaterialTheme.colors.error,
modifier = Modifier.padding(vertical = 16.dp)
            )
        }

        Button(
onClick = {
if (username.isNotEmpty() && password.isNotEmpty()) {
val user = databaseHelper.getUserByUsername(username)
if (user != null && user.password == password) {
                        error = "Successfully log in"
context.startActivity(
                            Intent(
                                context,
MainActivity::class.java
                            )
                        )
//onLoginSuccess()
}
else {
                        error =  "Invalid username or password"
}

                } else {
                    error = "Please fill all fields"
}
},
modifier = Modifier.padding(top = 16.dp)
        ) {
Text(text = "Login")
}
Row {
TextButton(onClick = {context.startActivity(
                Intent(
                    context,
RegisterActivity::class.java
                )
            )}
)
{ Text(text = "Register") }
TextButton(onClick = {
            })

{
Spacer(modifier = Modifier.width(60.dp))
                Text(text = "Forget password?")
}
        }
    }
}
private fun startMainPage(context: Context) {
val intent = Intent(context, MainActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
```

REGISTER ACTIVITY KT WITH DATABASE

```
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.background
import androidx.compose.foundation.layout.*
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat
//import com.example.owlapplication.ui.theme.OwlApplicationTheme

class RegisterActivity : ComponentActivity() {
private lateinit var databaseHelper: UserDatabaseHelper
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
databaseHelper = UserDatabaseHelper(this)
setContent {
RegistrationScreen(this, databaseHelper)
}
}
}

@Composable
fun RegistrationScreen(context: Context, databaseHelper:
UserDatabaseHelper) {

var username by remember { mutableStateOf("") }
var password by remember { mutableStateOf("") }
var email by remember { mutableStateOf("") }
var error by remember { mutableStateOf("") }

Column(
modifier = Modifier.fillMaxSize().background(Color.White),
horizontalAlignment = Alignment.CenterHorizontally,
verticalArrangement = Arrangement.Center
) {

Image(painterResource(id = R.drawable.study_signup), contentDescription =
"")

Text(
fontSize = 36.sp,
```

```
fontWeight = FontWeight.ExtraBold,
fontFamily = FontFamily.Cursive,
text = "Register"
)

Spacer(modifier = Modifier.height(10.dp))
TextField(
value = username,
onValueChange = { username = it },
label = { Text("Username") },
modifier = Modifier
                .padding(10.dp)
                .width(280.dp)

        )

TextField(
value = email,
onValueChange = { email = it },
label = { Text("Email") },
modifier = Modifier
                .padding(10.dp)
                .width(280.dp)
        )

TextField(
value = password,
onValueChange = { password = it },
label = { Text("Password") },
visualTransformation = PasswordVisualTransformation(),
modifier = Modifier
                .padding(10.dp)
                .width(280.dp)
        )


if (error.isNotEmpty()) {
Text(
text = error,
color = MaterialTheme.colors.error,
modifier = Modifier.padding(vertical = 16.dp)
            )
        }

Button(
onClick = {
if (username.isNotEmpty() && password.isNotEmpty() && email.isNotEmpty()) {
val user = User(
id = null,
firstName = username,
lastName = null,
email = email,
password = password
                    )
                    databaseHelper.insertUser(user)
                    error = "User registered successfully"
// Start LoginActivity using the current context
context.startActivity(
                        Intent(
                            context,
LoginActivity::class.java
```

```
)
                    )

                } else {
                    error = "Please fill all fields"
}
},
modifier = Modifier.padding(top = 16.dp)
        ) {
Text(text = "Register")
}
Spacer(modifier = Modifier.width(10.dp))
Spacer(modifier = Modifier.height(10.dp))

Row() {
Text(
modifier = Modifier.padding(top = 14.dp), text = "Have an account?"
)
TextButton(onClick = {
context.startActivity(
                    Intent(
                        context,
LoginActivity::class.java
)
                )
})

{
Spacer(modifier = Modifier.width(10.dp))
Text(text = "Log in")
}
        }
    }
}
private fun startLoginActivity(context: Context) {
val intent = Intent(context, LoginActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
```

MAIN ACTIVITY1.KT FILE

```
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
)
```

MAIN ACTIVITY2.KT FILE

```
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
)




}
}
```

MAIN ACTIVITY3.KT FILE

```
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

class MainActivity3 : ComponentActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContent {
Greeting1()
}
}
}
@Composable
fun Greeting1() {
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_2),
contentDescription = "",
modifier = Modifier.align(Alignment.CenterHorizontally)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )

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

Text(
text = stringResource(id = R.string.course2),
color = Color(0xFFFFA500),
fontSize = 16.sp,
modifier = Modifier.align(Alignment.CenterHorizontally)
        )

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

```
Text(
text = stringResource(id = R.string.topic2),
fontWeight = FontWeight.Bold,
fontSize = 26.sp,
modifier = Modifier.align(Alignment.CenterHorizontally)

        )
Spacer(modifier = Modifier.height(20.dp))
Text(
text = stringResource(id = R.string.subheading2_1),
modifier = Modifier.align(Alignment.Start),
fontSize = 20.sp
)

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

Text(
text = stringResource(id = R.string.text2_1),
modifier = Modifier.align(Alignment.Start),
textAlign = TextAlign.Justify,
fontSize = 16.sp
)

Spacer(modifier = Modifier.height(20.dp))
Text(
text = stringResource(id = R.string.subheading2_2),
modifier = Modifier.align(Alignment.Start),
fontSize = 20.sp
)

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

Text(
text = stringResource(id = R.string.text2_2),
modifier = Modifier.align(Alignment.Start),
textAlign = TextAlign.Justify,
fontSize = 16.sp
)




}
}
```

MAIN ACTIVITY4.KT FILE

```
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.MaterialTheme
import androidx.compose.material.Surface
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.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
//import com.example.owlapplication.ui.theme.Owl1ApplicationTheme

class MainActivity4 : ComponentActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContent {
Greeting2()
}
}
}
@Composable
fun Greeting2() {
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_3),
contentDescription = "",
modifier = Modifier.align(Alignment.CenterHorizontally)
                .scale(scaleX = 1.5F, scaleY = 2F)
        )

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

Text(
text = stringResource(id = R.string.course3),
color = Color(0xFFFFA500),
fontSize = 16.sp,
modifier = Modifier.align(Alignment.CenterHorizontally)
        )

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

Text(
text = stringResource(id = R.string.topic3),
fontWeight = FontWeight.Bold,
fontSize = 26.sp,
modifier = Modifier.align(Alignment.CenterHorizontally)

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

```
text = stringResource(id = R.string.subheading3_1),
modifier = Modifier.align(Alignment.Start),
fontSize = 20.sp
)

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

Text(
text = stringResource(id = R.string.text3_1),
modifier = Modifier.align(Alignment.Start),
textAlign = TextAlign.Justify,
fontSize = 16.sp
)

Spacer(modifier = Modifier.height(20.dp))
Text(
text = stringResource(id = R.string.subheading3_2),
modifier = Modifier.align(Alignment.Start),
fontSize = 20.sp
)

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

Text(
text = stringResource(id = R.string.text3_2),
modifier = Modifier.align(Alignment.Start),
textAlign = TextAlign.Justify,
fontSize = 16.sp
)




}
}
```

MAIN ACTIVITY5.KT FILE

```
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.MaterialTheme
import androidx.compose.material.Surface
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.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
//import com.example.owlapplication.ui.theme.OwlApplicationTheme

class MainActivity5: ComponentActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContent {
Greeting3()
}
}
}
@Composable
fun Greeting3() {
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_3),
contentDescription = "",
modifier = Modifier.align(Alignment.CenterHorizontally)
                .scale(scaleX = 1.5F, scaleY = 2F)
        )

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

Text(
text = stringResource(id = R.string.course4),
color = Color(0xFFFFA500),
fontSize = 16.sp,
modifier = Modifier.align(Alignment.CenterHorizontally)
        )

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

Text(
text = stringResource(id = R.string.topic4),
fontWeight = FontWeight.Bold,
fontSize = 26.sp,
modifier = Modifier.align(Alignment.CenterHorizontally)

        )
Spacer(modifier = Modifier.height(20.dp))
Text(
text = stringResource(id = R.string.subheading4_1),
modifier = Modifier.align(Alignment.Start),
fontSize = 20.sp
)

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

Text(
text = stringResource(id = R.string.text4_1),
modifier = Modifier.align(Alignment.Start),
```

```
textAlign = TextAlign.Justify,
fontSize = 16.sp
)

Spacer(modifier = Modifier.height(20.dp))
Text(
text = stringResource(id = R.string.subheading4_2),
modifier = Modifier.align(Alignment.Start),
fontSize = 20.sp
)

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

Text(
text = stringResource(id = R.string.text4_2),
modifier = Modifier.align(Alignment.Start),
textAlign = TextAlign.Justify,
fontSize = 16.sp
)




}
}
```

MODIFYING ANDROID MANIFEST

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

<application
android:allowBackup="true"
android:dataExtractionRules="@xml/data_extraction_rules"
android:fullBackupContent="@xml/backup_rules"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:supportsRtl="true"
android:theme="@style/Theme.OwlApplication"
tools:targetApi="31">

<activity
android:name=".RegisterActivity"
android:exported="false"
android:label="@string/title_activity_register"
android:theme="@style/Theme.OwlApplication" />
<activity
android:name=".MainActivity"
android:exported="false"
android:label="MainActivity"
android:theme="@style/Theme.OwlApplication" />
<activity
android:name=".MainActivity5"
android:exported="false"
android:label="@string/title_activity_main5"
android:theme="@style/Theme.OwlApplication" />
<activity
android:name=".MainActivity4"
```

```
android:exported="false"
android:label="@string/title_activity_main4"
android:theme="@style/Theme.OwlApplication" />
<activity
android:name=".MainActivity3"
android:exported="false"
android:label="@string/title_activity_main3"
android:theme="@style/Theme.OwlApplication" />
<activity
android:name=".MainActivity2"
android:exported="false"
android:label="@string/title_activity_main2"
android:theme="@style/Theme.OwlApplication" />
<activity
android:name=".LoginActivity"
android:exported="true"
android:label="@string/app_name"
android:theme="@style/Theme.OwlApplication">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>

</manifest>
```