PROJECT REPORT

1. 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.

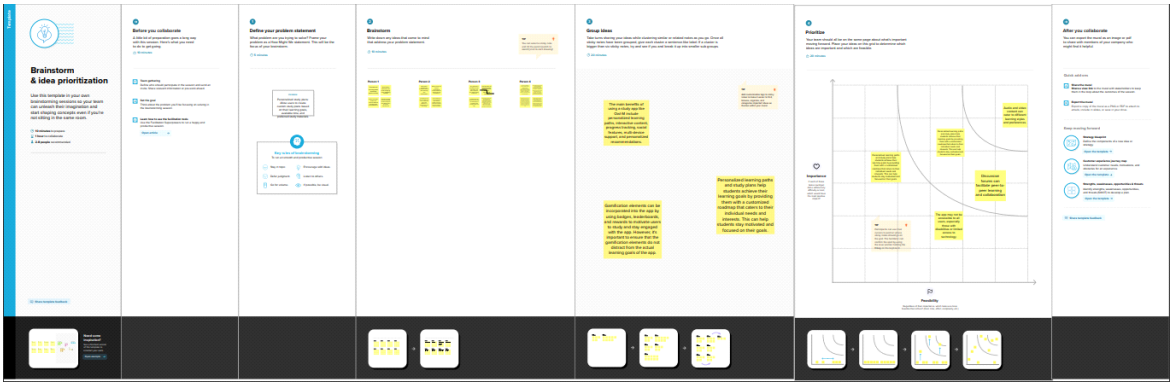
1.2 Purpose

- You′ll be able to work on Android studio and build an app.
- You′ll be able to integrate the database accordingly.

2. PROBLEM DEFINITION & DESIGN THINKING

2.1 Empathy Map

2.2 Ideation & Brainstorming Map



3. RESULT

Login Page :

Register Page :

Register

Username

Email

Password

Register

Have an account? Log in

MainPage :

Study Material

Arts & Craft

The Basics of Woodturning

Painting

An introduction to oil painting

Architecture

Book page :

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

The art on monuments in ancient Egypt offers

4. ADVANTAGES & DISADVANTAGES

4.1 Advantages

- User-Friendly Interface: The app has a simple and user-friendly interface that is easy to navigate and use. This makes it an ideal choice for students of all ages, including those who may not be tech-savvy.

- Customizable Study Sessions: Owl-M allows students to create custom study sessions that can be tailored to their individual learning needs. This means that students can focus on areas where they need the most help, ensuring that they retain information more effectively.
- Study Reminders: The app sends study reminders to users, helping them to stay on track with their studies. This is especially useful for students who have busy schedules and may struggle to find time to study.

4.2 Disadvantages

- Limited Content: The app has a limited number of pre-made flashcards, and users must create their own to study specific topics. This can be time-consuming and may not be ideal for students who are looking for a comprehensive study tool.
- No Audio or Video Support: The app does not support audio or video content, which can be a drawback for students who learn better through these mediums.

5. APPLICATIONS

- Study Schedule Management: Owl-M helps students to manage their study schedules effectively. The app allows users to create and organize their study plans, set reminders, and track their progress.

- Flashcard Creation: Owl-M enables users to create flashcards to aid their learning. Users can create flashcards with text, images, and audio to enhance their studying experience.
- Quiz Creation: The app allows users to create quizzes to test their knowledge. The quiz creation feature can be used to create different types of questions, such as multiple choice, true/false, and fill-in-the-blank questions.
- Progress Tracking: Owl-M provides users with the ability to track their progress through various learning modules. Users can see their progress in real-time and adjust their learning strategies accordingly.
- Material Design: Owl-M is built on material design principles, which means that the app is designed to be intuitive and user-friendly. The app's interface is clean and easy to navigate, making it an ideal tool for students of all ages.
- Collaboration: Owl-M allows users to collaborate with their peers, tutors, and teachers. Users can share study schedules, flashcards, and quizzes with others, making it easier to work together towards academic success.

6. CONCLUSION

Owl-M is an excellent material design study app that offers a range of features and benefits to users. Theapp provides students with a convenient and effective way to manage their study schedules, create flashcards, and track their progress. Its intuitive interface and material design principles make it easy to use and navigate, while its collaboration features allow users to work together towards academic success. Overall,

Owl-M is a powerful tool for students of all ages and backgrounds who want to improve their learning experience and achieve their academic goals.

7. FUTURE SCOPE

- Personalization: Owl-M could introduce more personalized features to cater to individual learning styles and preferences. This could include adaptive learning algorithms, personalized study plans, and customized quizzes.
- Augmented Reality: With the rise of augmented reality technology, Owl-M could incorporate AR features to enhance the learning experience. For example, users could visualize complex concepts in 3D or interact with virtual study aids.
- Integration with Other Learning Platforms: Owl-M could integrate with other learning platforms, such as online courses, textbooks, or e-learning tools. This could provide users with a more holistic and streamlined learning experience.
- Social Learning: Owl-M could expand its collaboration features to enable social learning, where users can connect with other learners and form study groups or communities.
- Gamification: The app could introduce more gamification elements to make studying more fun and engaging. This could include rewards systems, leaderboards, and mini-games.

8. APPENDIX

```
 //User.kt

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?,

 )


//UserDao.kt

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)
}


//UserDatabase.kt

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
 }
 }
 }
}


//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
 }

}

//LoginActivity.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.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)
}

//RegisterActivity.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.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)
}


//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
 )




 }
}

//MainActivity3.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

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
 )




 }
}


//MainActivity4.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.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 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
 )




 }
}
```

```
//MainActivity5.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.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 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
```

```
 )




 }
}


//AndroidManife.xml


<?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>
```