OWL-M:
A Material Design Study App

Project presented by:

TEAM ID:NM2023TMID09475

Team Leader: HARISH G

TEAM MEMBERS:

GUNASEELAN A

JIJENDRAN K

KALIYAPERUMAL V

SRIDHAR C

OWL-M: A MATERIAL DESIGN STUDY APP
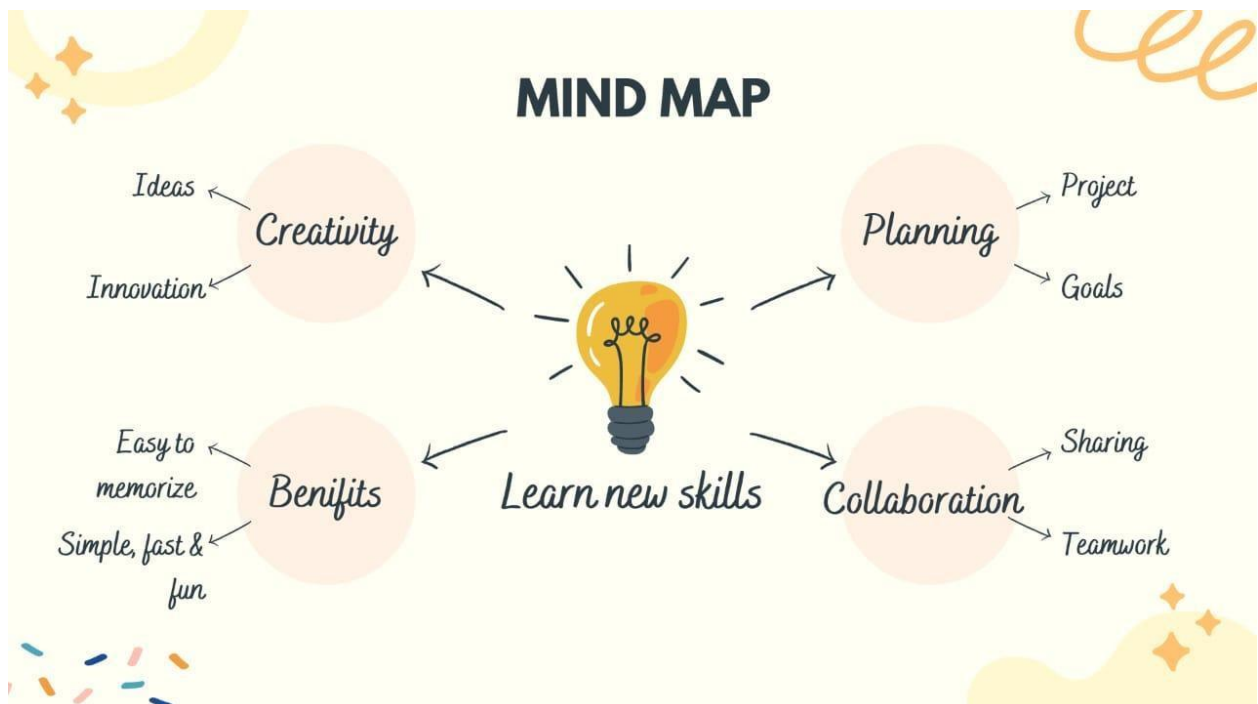
1 INTRODUCTION

1.1 Overview

This project is an Android implementation of Owl, a Material Study showcasing the possibilities of using Material Theming and Material Components for Android

1.2 Purpose

Owl is an educational app that uses Material Design components and Material Theming to create an energetic, motivational brand experience.
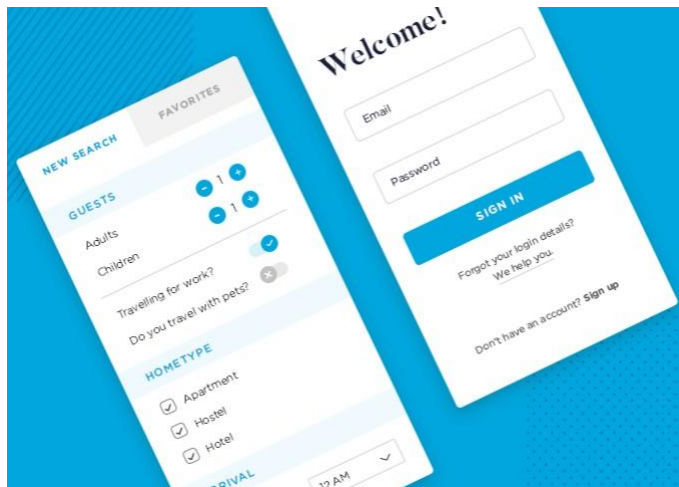
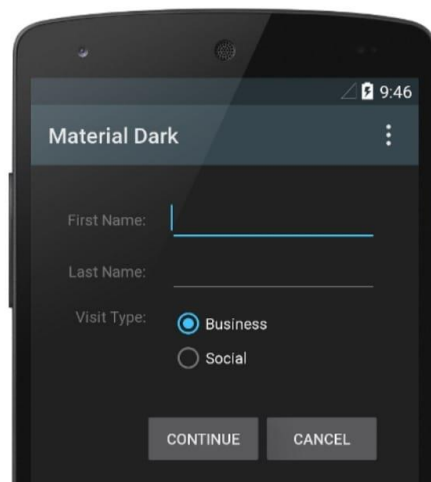## 2 Problem Definition & Design Thinking

2.1 Empathy Map

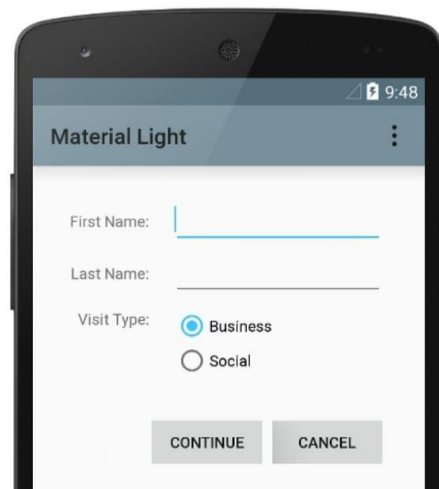## 2.2 Ideation & Brainstorming Map

To create a project mindmap that helps to stay on sechedule.

3 RESULT

Dark material
theme

Light material
theme

4 ADVANTAGES & DISADVANTAGES

Advantages

1.Responsive layout

It offers fast-scrolling screens and quick content updates, which are important for the mobile age.
2.Graceful animations

Material designs come with smooth, engaging, and flexible UI transitions and animations, making it easier for users to understand the relationship between different elements on their devices.

3. It has a learning curve

Many people don't like having lots of different things or actions that they need to remember when they access a site/app

 Disadvantages

1. Too many colors

Because the interface consists of many objects and different components, it becomes difficult for users to look at screens that contain too much color

2. The animations

 The objects which the material design uses move to different positions on the screen when a user performs an action such as scrolling down or clicking on an object

3. It has a learning curve

Many people don't like having lots of different things or actions that they need to remember when they access a site/app

5 APPLICATIONS

Create intuitive and beautiful products with Material Design

Material Design for Android

 Designing and Building Apps with Material Design

Material Design's Impact on Mobile App Design

6 CONCLUSION

Use Material studies to inspire your own adoption of Material Theming and components. These studies explore real-world design and product limitations through the examination of a set of fictional apps, each designed with unique properties and use cases.

7 FUTURE SCOPE

Materials planning and control: Material management and control can be explained in a scientific way that it is used to determine the needs of raw materials, components, and other commodities needed for production within the economic investment policies.

8 APPENDIX

A. Source Code

Attach the code for the solution built.

Owl-M : A Material Design Study App

AndroidManifest.xml

```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" />
```

```xml
        <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>
```

Color.kt

```kotlin
package com.example.owlapplication.ui.theme

import androidx.compose.ui.graphics.Color

val Purple200 = Color(0xFFBB86FC)
val Purple500 = Color(0xFF6200EE)
val Purple700 = Color(0xFF3700B3)
val Teal200 = Color(0xFF03DAC5)
```

Shape.kt

```kotlin
package com.example.owlapplication.ui.theme

import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material.Shapes
```

```
import androidx.compose.ui.unit.dp

val Shapes = Shapes(
    small = RoundedCornerShape(4.dp),
    medium = RoundedCornerShape(4.dp),
    large = RoundedCornerShape(0.dp)
)
```

Theme.kt

```
package com.example.owlapplication.ui.theme

import androidx.compose.foundation.isSystemInDarkTheme
import androidx.compose.material.MaterialTheme
import androidx.compose.material.darkColors
import androidx.compose.material.lightColors
import androidx.compose.runtime.Composable

private val DarkColorPalette = darkColors(
    primary = Purple200,
    primaryVariant = Purple700,
    secondary = Teal200
)

private val LightColorPalette = lightColors(
    primary = Purple500,
    primaryVariant = Purple700,
    secondary = Teal200

    /* Other default colors to override
    background = Color.White,
    surface = Color.White,
    onPrimary = Color.White,
    onSecondary = Color.Black,
    onBackground = Color.Black,
    onSurface = Color.Black,
    */
)

@Composable
fun OwlApplicationTheme(
    darkTheme: Boolean = isSystemInDarkTheme(),
    content: @Composable () -> Unit
) {
    val colors = if (darkTheme) {
        DarkColorPalette
    } else {
        LightColorPalette
    }

    MaterialTheme(
        colors = colors,
        typography = Typography,
```

```
        shapes = Shapes,
        content = content
    )
}
```

Type.kt

```
package com.example.owlapplication.ui.theme

import androidx.compose.material.Typography
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.sp

// Set of Material typography styles to start with
val Typography = Typography(
    body1 = TextStyle(
        fontFamily = FontFamily.Default,
        fontWeight = FontWeight.Normal,
        fontSize = 16.sp
    )
    /* Other default text styles to override
    button = TextStyle(
        fontFamily = FontFamily.Default,
        fontWeight = FontWeight.W500,
        fontSize = 14.sp
    ),
    caption = TextStyle(
        fontFamily = FontFamily.Default,
        fontWeight = FontWeight.Normal,
        fontSize = 12.sp
    )
    */
)
```

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
```

```kotlin
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(
```

```kotlin
        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)
}
```

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)
```

```kotlin
            .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()
```

```kotlin
                .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
```

```kotlin
                                    .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

```kotlin
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 = "",
```

```kotlin
    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(0xFFFFFA500),
    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
)
```

```
        }
}



```

MainActivity2.kt

```kotlin
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)
        )
```

```kotlin
        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
        )



    }
}
```

MainActivity3.kt

```kotlin
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))
```

```kotlin
        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

```kotlin
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))
```

```kotlin
        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

```kotlin
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_4),
            contentDescription = "",
            modifier = Modifier.align(Alignment.CenterHorizontally)
                .scale(scaleX = 1.5F, scaleY = 1.5F)
        )

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

```kotlin
            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
            )




    }
}
```

RegisterActivity.kt

```kotlin
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(
```

```kotlin
            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
                    )
```

```kotlin
                    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)
}
```

User.kt

```kotlin
package com.example.owlapplication

import androidx.room.ColumnInfo
import androidx.room.Entity
```

```kotlin
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

```kotlin
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

```kotlin
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 {
```

```kotlin
            @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

```kotlin
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)
```

```kotlin
    }

    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 =
```

```kotlin
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
    }

}
```