A MATERIAL DESIGN STUDY APP

INTRODUCTION

1.1 OVERVIEW

1.2 Project Description :

A project that demonstrates the use of Android Jetpack compose to built a Ul for a Owl-M: a material design study app. Ow-M app is a sampe projet built using the Android Compose U toolkit. A Compose implementation of the Owl Material study.

Learning Outcomes :

By end of this project:

- Will be able to work on Android study and built an app.
- will be able to integrate the database accordingly.

Project Workflow :

- Users register into the application.
- After registration, user logins into the application.
- User enters into the main page
- User can view the subject themes on selecting theme he can read about it.

1.2 PURPOSE

Owl

Owl is an educational app that uses Material Design components and Material Theming to create an energetic, motivational brand experience.

About Owl

Owl is an educational app that provides courses for people who what to explore and learn skis in design, art, architecture, and fashion. The Owl brand uses bold color, shape, and typography to express its brand attributes: energy, daring, and fun.

Bold aesthetic

Owl's design reflects the energy and excitement of learning a new skill, using a bold aesthetic that expresses exploration and growth. Its Ul contains unfiled shapes that invite the user to populate them with new content and coutses. Its copywriting speaks with an inviting, can-do tone of voice.

Layout

Grid system

Owl uses a responsive grid system, which has flexible columns and padding that can resize depending on the screen width (such as mobile, tablet, or desktop).

Color

Owl has three primary colors. Each color is used to create a distinct visual theme for each section.

Typography

Type scale

Owl's type scale provides the typography variety necessary for the app content. All iteams in the type scale use Rubik as the typeface, and make use of the veriety of weights available by using Rubik Reguler, Medium, and Bold.

Rubik

Owl uses the Rubi typeface. Rubik is sans serif, with slightly rounded corners, and has many font weights.

Iconography

Owl's Iconography reflects the curved shape and style of it's brand typeface Rubik.

Shape

Categories

Components are grouped into shape categories based on their size. Shape categories let you set multiple components vaues at once. Shape categories include.

Small components

Exapanding bottom sheet are in the small shape category. In their collapsed stste, they have a rounded corner with a 50% corner radiuls.

Medium components

Cords are in the medium shape category. Their corner are rounded, with a 0dp corner redius.

Large components

Side sheets are in the large shape category. The corner on their leading side are rounded, with a 16dp cornet radius.

Components

lists and Dividers

Owl uses the list components in a variety of ways.

This list shows result, using custom typography, color, iconography, and imagery. Inset dividers separate items.

Cards

Owl featured courses are presented using a card collection. They have custom shape, color, typography, and iconography, with center-aligned content. There is little padding between cards, and the collection uses a masonry pattern.

Sheets

Owl uses sheets to present course tutorial videos. On desktop and tablet, these videos can be fount in a side sheet customized with the Learn section's primary color: pink.

Bottom navigation

Owl's bottom navigation components has custom color, typography, and iconography.

Motion

Launch screen

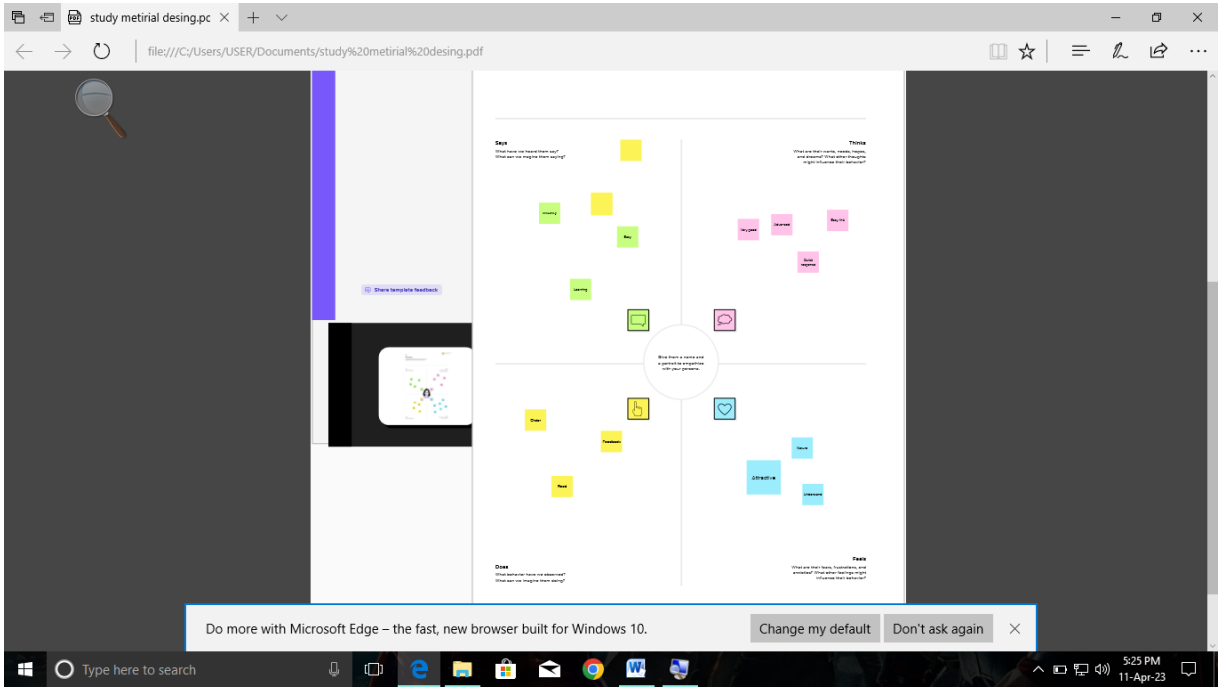
Owl's launch screen display a character animation to give the app a playful quality.

Animated icons

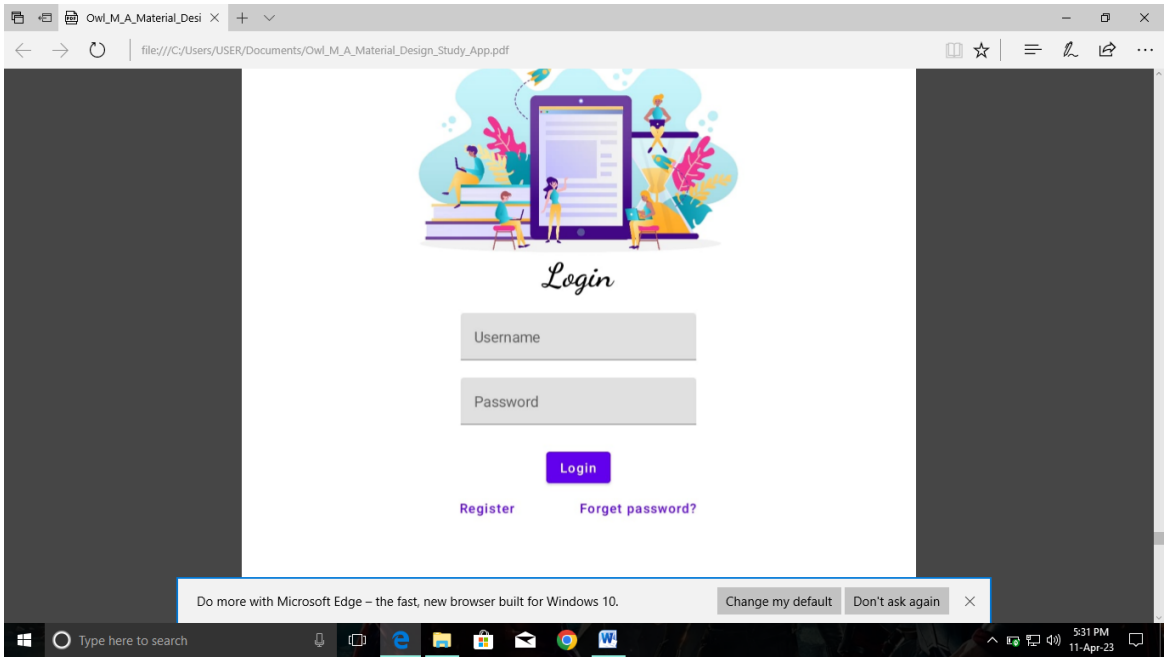
Owl's icon animations are inspired by both its logo abnd loading state animations.

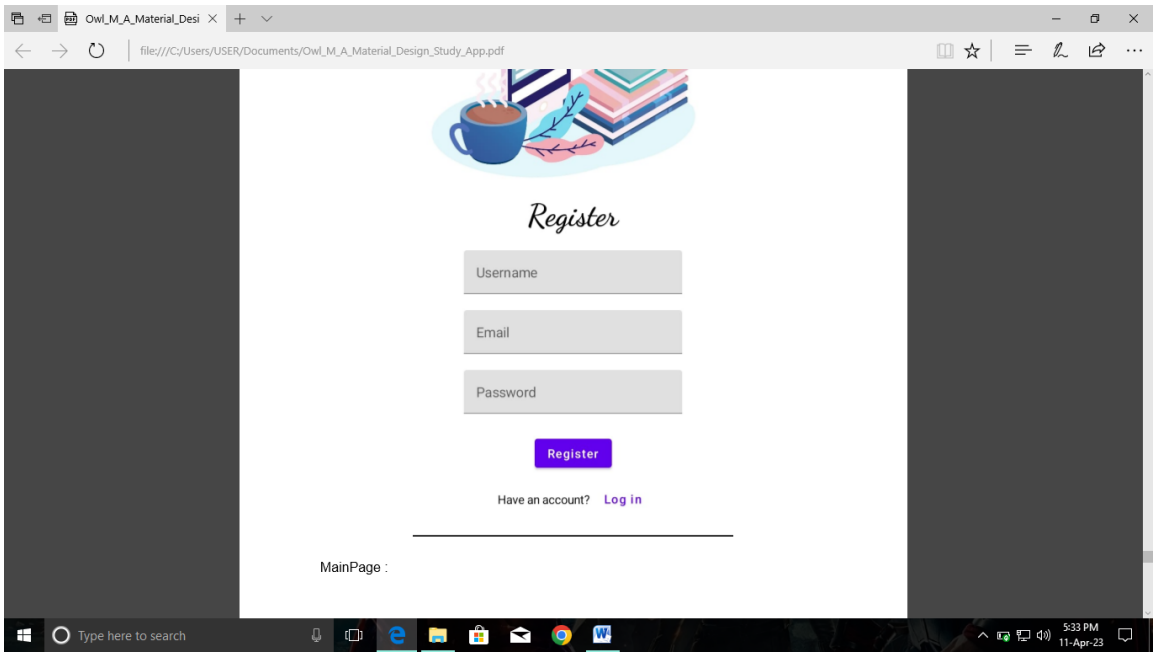
Problem Definition & Design Thinking

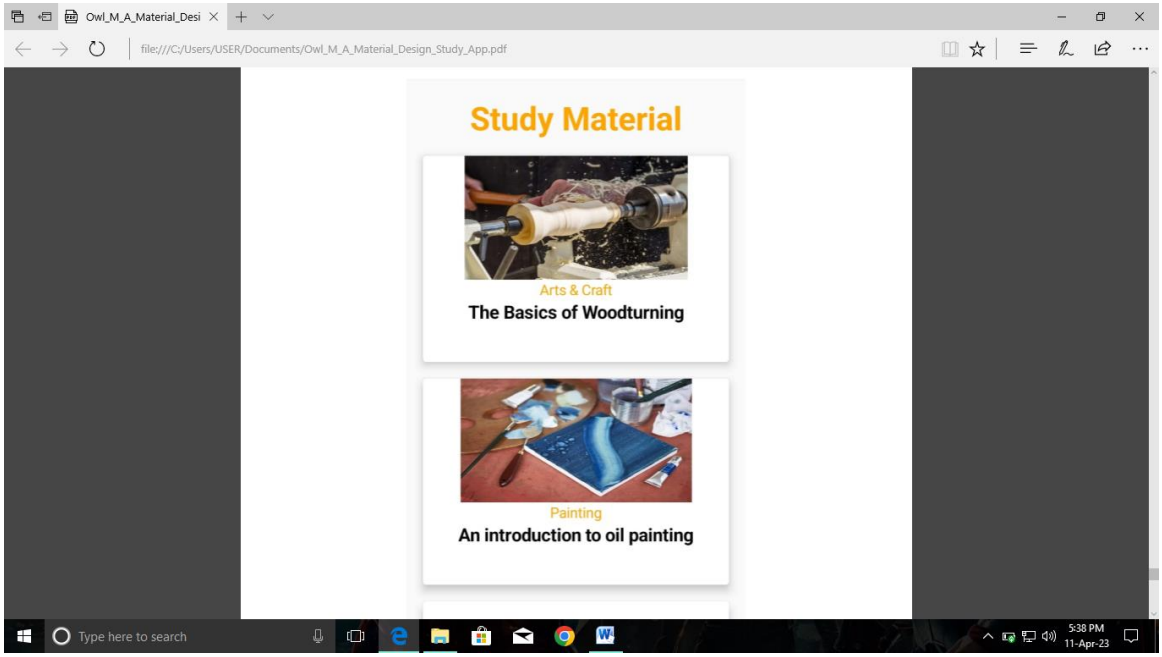
2.1 Empathy Map

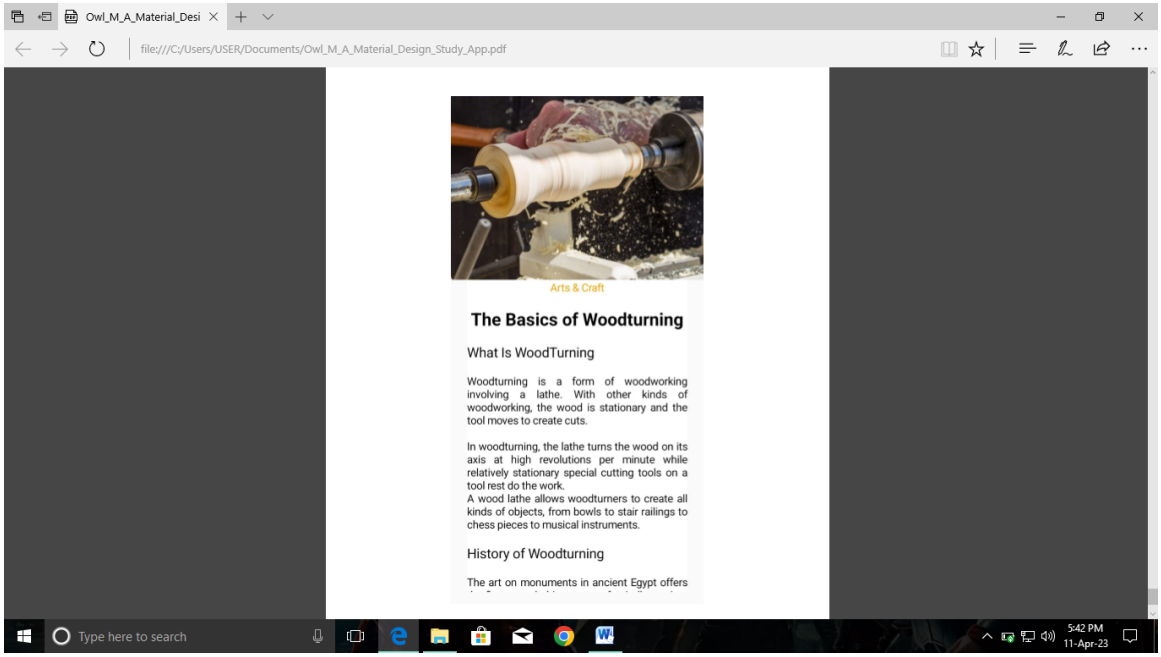


RESULT


Owl_M_A_Material_Desi
file:///C:/Users/USER/Documents/Owl_M_A_Material_Design_Study_App.pdf
Login
Username
Password
Login
Register
Forget password?
Do more with Microsoft Edge – the fast, new browser built for Windows 10.
Change my default
Don't ask again
Type here to search
5:31 PM
11-Apr-23


Owl_M_A_Material_Desi
file:///C:/Users/USER/Documents/Owl_M_A_Material_Design_Study_App.pdf
Register
Username
Email
Password
Register
Have an account? Log in
MainPage :
Type here to search
5:33 PM
11-Apr-23


Owl_M_A_Material_Desi
file:///C:/Users/USER/Documents/Owl_M_A_Material_Design_Study_App.pdf
Study Material
Arts & Craft
The Basics of Woodturning
Painting
An introduction to oil painting
Type here to search
5:38 PM
11-Apr-23


Owl_M_A_Material_Desi
file:///C:/Users/USER/Documents/Owl_M_A_Material_Design_Study_App.pdf
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
Type here to search
5:42 PM
11-Apr-23

ADVANTAGE

As with any well-established design system, there are some major pros to using a Material Design system that designers should consider.

Google's Material Design is effectively an entire design ecosystem, rather than just a set of style guidelines. If there's a potential design situation that exists, Material Design likely has a comprehensive set of rules for how to handle it. That includes complex use cases that are often overlooked by less comprehensive design systems. This can be very comforting for designers who want that kind of structure.

Google maintains Material Design and keeps extensive documentation for how to use and implement it. This kind of support and documentation can be lacking for many modern design systems.

Despite all this comprehensiveness and documentation, Material Design remains a fairly flexible design library. Within the guidelines, much of the specifics of how to implement the design are left entirely up to the designer.

More granular advantages for Material Design include things like subtle skeuomorphism, which sets it apart from flat design and makes it more intuitive for many users. Another user-friendly feature is that user feedback in the form of haptic feedback, subtle animations, and similar things are built into the guidelines. It has a very simplified sense of physics, too, which makes interactions more intuitive.

Material Design was built on a mobile-first sensibility, which makes sense considering its original purpose was for designing Android apps. It also

promotes animation in designs, both for user feedback and to hint at how different features function.

Finally, dark theme options have been made available, adding even more visual flexibility for designers. Originally, Material Design was very light and bright, which didn’t work well with the aesthetics of some brands. The addition of a dark theme guideline fixes that issue.

DISADVANTAGE

While Material Design has very obvious pros, that doesn't mean there aren't cons that go along with using it.

First up, Material Design is immediately identifiable and is strongly associated with Google and, specifically, Android. While this isn't necessarily a bad thing for everyone, it's potentially a negative for some.

One big reason that it might be a negative is that it limits the effectiveness of other branding while using the Google design system. Yes, designers can incorporate logos, color palettes (within the Material Design guidelines), and other differentiating factors to support the brand identity, but a product following the Material Design specifications will almost always *also* be associated with Google.

Since motion and animation are promoted within the Material Design guidelines, sites or apps that don't incorporate it can seem to users as if they're missing something. People associate the motion characteristics of Material Design with the visual characteristics, which can leave designs without motion lacking.

Sure, one solution is to always incorporate motion in designs that follow the Material Design specs. But extensive animations can be very resource-heavy on mobile devices, resulting in higher data usage and faster battery depletion. It's a balancing act designers have to consider when working within the Material Design guidelines.

Beginners may find that the Material Design specification is more complicated and harder to implement than other styles like flat design. Because the Material Design system is so

comprehensive, there are a lot more things to consider and adhere to than many new designers may be comfortable with.

Its comprehensiveness can also lead to some designers feeling constrained and unable to fully realize their own creativity. It can also stifle innovation, since virtually any design challenge has been planned for and solutions offered. While being helpful in many cases, this can prevent designers from taking new approaches to problems, while also limiting the number of new ideas that may occur.

There are also some usability issues in Material Design for websites and apps that can make them very user-*un*friendly. One of the biggest issues is with the so-called "mystery meat" navigation encountered on many mobile design apps. Icons are often used in place of text, and while sometimes the icons are immediately recognizable and fairly usable, at other times they're not.

A circle to indicate "Home" is significantly harder to identify than the house icon that was previously used in most Android interfaces. This is a prime example of placing form over function, which is a holdover from Material Design's flat design roots.

And it's not just in the lower navigation bar. Material Design's preference for including circular floating action buttons is also a usability issue. These circular buttons only include space for an icon, with no assistive text included. And because icons can be so open to interpretation, in many cases, users are left questioning what these buttons actually do.

APPICATION

- Knowledge augmentation
- Taiored earning experiences
- Improved engagement
- Access to online study material
- Ease of communication
- Most significantly.

Conclusion

Material Design is a great concept. It create consistent user experiences and simplifies website and app design, ensring all users will easiy access buttons and layouts. While no design system is perfect, having common standards that will reduce bad experiences is always a good idears.

APPENDIX

User.kt

```
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?,
    )
```

UserDeo.kt

```
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)
```

```
}
```

RegisterActivity.kt

```
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)
}
```

UserDatabaseHelper.kt

es (101 sloc) 4.22 KB

Raw Blame

```
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.owlapplic
ation

                        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.PasswordVisualTran
sformation

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.OwlApplicatio
nTheme

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.owlapplic
ation

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.PasswordVisualTran
sformation
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.OwlApplicatio
nTheme
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.owlapplicatio
n

    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.rememberScrollSta
    te
    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.owlapplica
tion

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.OwlApplicati
```

```
onTheme
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.owlapplicatio
n

  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.rememberScrollSta
  te
```

```
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
```

```
                )
            }
        }
```

MainActivity5.kt

```
package
com.example.owlapplica
tion
                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.OwlApplicati
onTheme
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.owlapplica
tion

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.OwlApplicati
onTheme
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
                        )
                    }
                    }
```

AndroidManifest.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>
```

https://github.com/smartinternz02/Owl-M-A-Material-Design-Study-App