

# *Owl-M : A Material Design Study App*

Presented By

**Team ID: NM2023TMID34995**

**Team Leader: S.ARIKESH**

**Team members:**

**S.SHASTI PRASATH**

**S. JUDITH JOSHWA**

**M.MOHAMED ISRATH**

**R.PRAVEEN KUMAR**

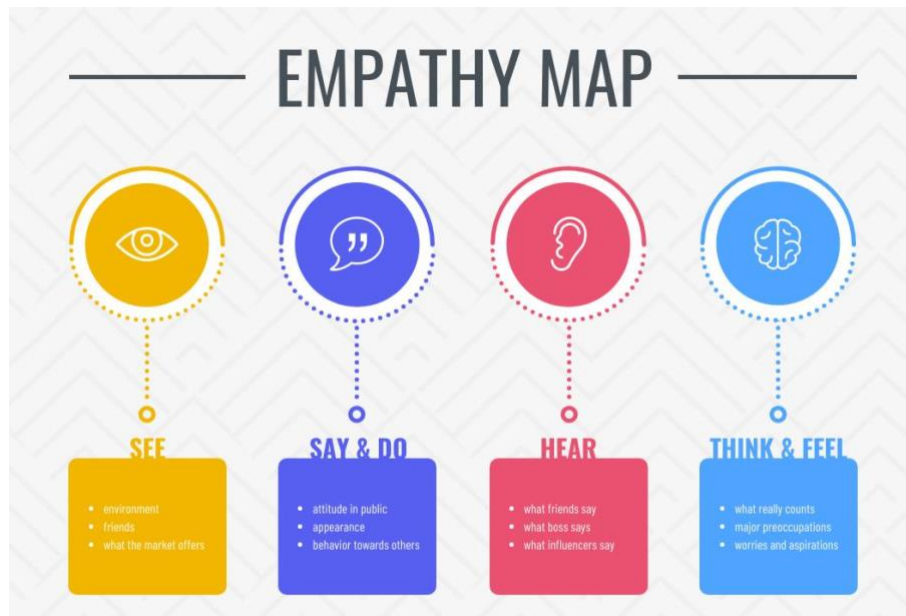
## **1.INDRODUCTION:**

Owl-M is a modern and sleek study app designed with Material Design principles in mind. It provides a seamless and intuitive user experience to help students stay organized and focused on their studies. With its clean and simple interface, Owl-M makes it easy for users to create and manage study schedules, set reminders, take notes, and track their progress.

The app is designed to be both functional and visually appealing, with a range of customizable themes and color schemes to choose from. Its minimalist design ensures that users can quickly find what they need and stay on task without unnecessary distractions.

Whether you're a student looking for a reliable study companion or someone who wants to stay on top of their personal learning goals, Owl-M is the perfect app for you. It offers all the features you need to stay organized, motivated, and successful in your studies. So why wait? Download Owl-M today and take the first step towards achieving your learning goals!

## Empathy map:



## Brainstorm Map:



**Advantages :**

1. User-Friendly Interface: Owl-M's interface is clean, easy to navigate, and aesthetically pleasing, making it easy for users to use and stay engaged.
2. Customizable Study Plans: Owl-M allows users to create personalized study plans, making it easier to track progress and stay motivated.
3. Multiple Study Modes: Owl-M offers multiple study modes, including flashcards, quizzes, and summaries, allowing users to tailor their studying experience to their individual needs.
4. Integration with Other Tools: Owl-M seamlessly integrates with other productivity tools, such as Google Drive and Google Calendar, making it easy to stay organized and on track.
5. Sync Across Devices: Owl-M syncs user data across devices, so users can easily switch between studying on their phone, tablet, or computer.

**Disadvantages :**

1. Limited content: If the app has a limited number of study materials or courses, users may quickly exhaust the available content and lose interest in the app.
2. Technical issues: Any app can encounter technical issues, such as bugs, crashes, or slow performance, which can negatively affect the user experience.
3. Poor user interface: A confusing or cluttered user interface can make it difficult for users to navigate the app and find the features they need.
4. Lack of personalization: Users may be looking for an app that can cater to their specific needs and preferences, and if Owl-M lacks personalization options, it may not meet their expectations.

5. Cost: If the app requires a subscription or has in-app purchases, users may be hesitant to invest in it, especially if they can find similar content for free elsewhere

### **Future scope:**

Overall, the future scope for Owl-M is promising, and there are many opportunities for growth and development that could help the app continue to be a valuable resource for designers and creatives.

### **Appendix:**

Detailed guides that explain how to use different features of the app, such as how to navigate the interface, access study materials, and track progress.

### **SOURCE CODE:**

```
<?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>
```

Color.kt

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

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

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,
            onChange = { 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)
}

```

MainActivity.kt

Package com.example.owlapplication

```
Import android.content.Context
Import android.content.Intent
Import android.os.Bundle
Import androidx.activity.ComponentActivity
Import androidx.activity.compose.setContent
Import androidx.compose.foundation.Image
Import androidx.compose.foundation.clickable
Import androidx.compose.foundation.layout.*
Import androidx.compose.foundation.rememberScrollState
Import androidx.compose.foundation.verticalScroll
Import androidx.compose.material.Card
Import androidx.compose.material.Text
Import androidx.compose.runtime.Composable
Import androidx.compose.ui.Alignment
Import androidx.compose.ui.Modifier
Import androidx.compose.ui.draw.scale
Import androidx.compose.ui.graphics.Color
Import androidx.compose.ui.res.painterResource
Import androidx.compose.ui.res.stringResource
Import androidx.compose.ui.text.font.FontWeight
Import androidx.compose.ui.text.style.TextAlign
Import androidx.compose.ui.unit.dp
Import androidx.compose.ui.unit.sp
```

```
Class MainActivity : ComponentActivity() {
    Override fun onCreate(savedInstanceState: Bundle?) {
        Super.onCreate(savedInstanceState)
        setContent {
            StudyApp(this)
        }
    }
}
```

@Composable

```

Fun StudyApp(context: Context) {

    Column(
        Modifier = Modifier
            .padding(20.dp)
            .verticalScroll(rememberScrollState())

    ) {

        Text(text = "Study Material",
            fontSize = 36.sp,
            fontWeight = FontWeight.Bold,
            color = Color(0xFFFFFA500),
            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(0xFFFFFA500),
            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(0xFFFFFA500),
        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(0xFFFFFA500),
                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(0xFFFFFA500),
            fontSize = 16.sp)

        Text(
            Text = stringResource(id = R.string.topic4),
            fontWeight = FontWeight.Bold,
            fontSize = 20.sp,
            textAlign = TextAlign.Center,
        )
    }
}

}
}

```

MainActivity2.kt

Package com.example.owlapplication

```
Import android.os.Bundle
Import androidx.activity.ComponentActivity
Import androidx.activity.compose.setContent
Import androidx.compose.foundation.Image
Import androidx.compose.foundation.background
Import androidx.compose.foundation.layout.*
Import androidx.compose.foundation.rememberScrollState
Import androidx.compose.foundation.verticalScroll
Import androidx.compose.material.Text
Import androidx.compose.runtime.Composable
Import androidx.compose.ui.Alignment
Import androidx.compose.ui.Modifier
Import androidx.compose.ui.draw.scale
Import androidx.compose.ui.graphics.Color
Import androidx.compose.ui.res.painterResource
Import androidx.compose.ui.res.stringResource
Import androidx.compose.ui.text.font.FontWeight
Import androidx.compose.ui.text.style.TextAlign
Import androidx.compose.ui.unit.dp
Import androidx.compose.ui.unit.sp
Import com.example.owlapplication.ui.theme.OwlApplicationTheme
```

```
Class MainActivity2 : ComponentActivity() {
    Override fun onCreate(savedInstanceState: Bundle?) {
        Super.onCreate(savedInstanceState)
        setContent {
            Greeting()
        }
    }
}
@Composable
Fun Greeting() {
    Column(
```

```
Modifier = Modifier.padding(start = 26.dp, end = 26.dp, bottom = 26.dp)
    .verticalScroll(rememberScrollState())
    .background(Color.White),
verticalArrangement = Arrangement.Top
){
```

```
Image(
    painterResource(id = R.drawable.img_1),
    contentDescription = "",
    modifier = Modifier.align(Alignment.CenterHorizontally)
    .scale(scaleX = 1.5F, scaleY = 1.5F)
)
```

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

```
Text(
    Text = stringResource(id = R.string.course1),
    Color = Color(0xFFFFA500),
    fontSize = 16.sp,
    modifier = Modifier.align(Alignment.CenterHorizontally)
)
```

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

```
Text(
    Text = stringResource(id = R.string.topic1),
    fontWeight = FontWeight.Bold,
    fontSize = 26.sp,
    modifier = Modifier.align(Alignment.CenterHorizontally)
)
```

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

```
Text(
    Text = stringResource(id = R.string.subheading1_1),
    Modifier = Modifier.align(Alignment.Start),
```

```
        fontSize = 20.sp  
    )
```

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

```
    Text(  
        Text = stringResource(id = R.string.text1_1),  
        Modifier = Modifier.align(Alignment.Start),  
        textAlign = TextAlign.Justify,  
        fontSize = 16.sp  
    )
```

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

```
    Text(  
        Text = stringResource(id = R.string.subheading1_2),  
        Modifier = Modifier.align(Alignment.Start),  
        fontSize = 20.sp  
    )
```

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

```
    Text(  
        Text = stringResource(id = R.string.text1_2),  
        Modifier = Modifier.align(Alignment.Start),  
        textAlign = TextAlign.Justify,  
        fontSize = 16.sp  
    )
```

```
    }  
}
```

MainActivity2.kt

Package com.example.owlapplication

```
Import android.os.Bundle
Import androidx.activity.ComponentActivity
Import androidx.activity.compose.setContent
Import androidx.compose.foundation.Image
Import androidx.compose.foundation.background
Import androidx.compose.foundation.layout.*
Import androidx.compose.foundation.rememberScrollState
Import androidx.compose.foundation.verticalScroll
Import androidx.compose.material.Text
Import androidx.compose.runtime.Composable
Import androidx.compose.ui.Alignment
Import androidx.compose.ui.Modifier
Import androidx.compose.ui.draw.scale
Import androidx.compose.ui.graphics.Color
Import androidx.compose.ui.res.painterResource
Import androidx.compose.ui.res.stringResource
Import androidx.compose.ui.text.font.FontWeight
Import androidx.compose.ui.text.style.TextAlign
Import androidx.compose.ui.unit.dp
Import androidx.compose.ui.unit.sp
```

```
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
)
```

}

}

MainActivity3.kt

Package com.example.owlapplication

```
Import android.os.Bundle
Import androidx.activity.ComponentActivity
Import androidx.activity.compose.setContent
Import androidx.compose.foundation.Image
Import androidx.compose.foundation.background
Import androidx.compose.foundation.layout.*
Import androidx.compose.foundation.rememberScrollState
Import androidx.compose.foundation.verticalScroll
Import androidx.compose.material.Text
Import androidx.compose.runtime.Composable
Import androidx.compose.ui.Alignment
Import androidx.compose.ui.Modifier
Import androidx.compose.ui.draw.scale
Import androidx.compose.ui.graphics.Color
Import androidx.compose.ui.res.painterResource
Import androidx.compose.ui.res.stringResource
Import androidx.compose.ui.text.font.FontWeight
Import androidx.compose.ui.text.style.TextAlign
Import androidx.compose.ui.unit.dp
Import androidx.compose.ui.unit.sp
```

```
Class MainActivity3 : ComponentActivity() {
    Override fun onCreate(savedInstanceState: Bundle?) {
        Super.onCreate(savedInstanceState)
        setContent {
            Greeting1()
        }
    }
}
@Composable
Fun Greeting1() {
    Column(
        Modifier = Modifier.padding(start = 26.dp, end = 26.dp, bottom = 26.dp)
        .verticalScroll(rememberScrollState())
    )
}
```

```

        .background(Color.White),
        verticalArrangement = Arrangement.Top
    ){

        Image(
            painterResource(id = R.drawable.img_2),
            contentDescription = "",
            modifier = Modifier.align(Alignment.CenterHorizontally)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )

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

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

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

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

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

```

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

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

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

```
Text(  
    Text = stringResource(id = R.string.subheading2_2),  
    Modifier = Modifier.align(Alignment.Start),  
    fontSize = 20.sp  
)
```

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

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

```
    }  
}
```

MainActivity4.kt

Package com.example.owlapplication

```
Import android.os.Bundle
Import androidx.activity.ComponentActivity
Import androidx.activity.compose.setContent
Import androidx.compose.foundation.Image
Import androidx.compose.foundation.background
Import androidx.compose.foundation.layout.*
Import androidx.compose.foundation.rememberScrollState
Import androidx.compose.foundation.verticalScroll
Import androidx.compose.material.MaterialTheme
Import androidx.compose.material.Surface
Import androidx.compose.material.Text
Import androidx.compose.runtime.Composable
Import androidx.compose.ui.Alignment
Import androidx.compose.ui.Modifier
Import androidx.compose.ui.draw.scale
Import androidx.compose.ui.graphics.Color
Import androidx.compose.ui.res.painterResource
Import androidx.compose.ui.res.stringResource
Import androidx.compose.ui.text.font.FontWeight
Import androidx.compose.ui.text.style.TextAlign
Import androidx.compose.ui.tooling.preview.Preview
Import androidx.compose.ui.unit.dp
Import androidx.compose.ui.unit.sp
Import com.example.owlapplication.ui.theme.OwlApplicationTheme
```

```
Class MainActivity4 : ComponentActivity() {
    Override fun onCreate(savedInstanceState: Bundle?) {
        Super.onCreate(savedInstanceState)
        setContent {
            Greeting2()
        }
    }
}
@Composable
Fun Greeting2() {
```

```
Column(  
    Modifier = Modifier.padding(start = 26.dp, end = 26.dp, bottom = 26.dp)  
        .verticalScroll(rememberScrollState())  
        .background(Color.White),  
    verticalArrangement = Arrangement.Top  
) {
```

```
    Image(  
        painterResource(id = R.drawable.img_3),  
        contentDescription = "",  
        modifier = Modifier.align(Alignment.CenterHorizontally)  
            .scale(scaleX = 1.5F, scaleY = 2F)  
    )
```

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

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

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

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

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

```
    Text(  
        Text = stringResource(id = R.string.subheading3_1),
```

```
        Modifier = Modifier.align(Alignment.Start),
        fontSize = 20.sp
    )
```

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

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

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

```
Text(
    Text = stringResource(id = R.string.subheading3_2),
    Modifier = Modifier.align(Alignment.Start),
    fontSize = 20.sp
)
```

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

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

```
    }
}
```

MainActivity5.kt

Package com.example.owlapplication

Import android.os.Bundle

Import androidx.activity.ComponentActivity

Import androidx.activity.compose.setContent

Import androidx.compose.foundation.Image

Import androidx.compose.foundation.background

Import androidx.compose.foundation.layout.\*

Import androidx.compose.foundation.rememberScrollState

Import androidx.compose.foundation.verticalScroll

Import androidx.compose.material.MaterialTheme

Import androidx.compose.material.Surface

Import androidx.compose.material.Text

Import androidx.compose.runtime.Composable

Import androidx.compose.ui.Alignment

Import androidx.compose.ui.Modifier

Import androidx.compose.ui.draw.scale

Import androidx.compose.ui.graphics.Color

Import androidx.compose.ui.res.painterResource

Import androidx.compose.ui.res.stringResource

Import androidx.compose.ui.text.font.FontWeight

Import androidx.compose.ui.text.style.TextAlign

Import androidx.compose.ui.tooling.preview.Preview

Import androidx.compose.ui.unit.dp

Import androidx.compose.ui.unit.sp

Import com.example.owlapplication.ui.theme.OwlApplicationTheme

Class 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))

        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  
Package com.example.owlapplication
```

```
Import android.content.Context  
Import android.content.Intent  
Import android.os.Bundle  
Import androidx.activity.ComponentActivity  
Import androidx.activity.compose.setContent  
Import androidx.compose.foundation.Image  
Import androidx.compose.foundation.background  
Import androidx.compose.foundation.layout.*  
Import androidx.compose.material.*  
Import androidx.compose.runtime.*  
Import androidx.compose.ui.Alignment  
Import androidx.compose.ui.Modifier  
Import androidx.compose.ui.graphics.Color  
Import androidx.compose.ui.layout.ContentScale  
Import androidx.compose.ui.res.painterResource  
Import androidx.compose.ui.text.font.FontFamily  
Import androidx.compose.ui.text.font.FontWeight  
Import androidx.compose.ui.text.input.PasswordVisualTransformation  
Import androidx.compose.ui.tooling.preview.Preview  
Import androidx.compose.ui.unit.dp  
Import androidx.compose.ui.unit.sp  
Import androidx.core.content.ContextCompat  
Import com.example.owlapplication.ui.theme.OwlApplicationTheme
```

```
Class RegisterActivity : ComponentActivity() {  
    Private lateinit var databaseHelper: UserDatabaseHelper  
    Override fun onCreate(savedInstanceState: Bundle?) {  
        Super.onCreate(savedInstanceState)  
        databaseHelper = UserDatabaseHelper(this)  
        setContent {
```

```

        RegistrationScreen(this, databaseHelper)
    }
}
}

```

@Composable

```

Fun RegistrationScreen(context: Context, databaseHelper: UserDatabaseHelper) {

```

```

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

```

```

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

```

```

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

```

```

    Text(
        fontSize = 36.sp,
        fontWeight = FontWeight.ExtraBold,
        fontFamily = FontFamily.Cursive,
        text = "Register"
    )

```

```

    Spacer(modifier = Modifier.height(10.dp))

```

```

    TextField(
        Value = username,
        onChange = { 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)
}

```

User.kt

Package com.example.owlapplication

Import androidx.room.ColumnInfo

Import androidx.room.Entity

Import androidx.room.PrimaryKey

@Entity(tableName = "user\_table")

Data class User(

@PrimaryKey(autoGenerate = true) val id: Int?,

@ColumnInfo(name = "first\_name") val firstName: String?,

@ColumnInfo(name = "last\_name") val lastName: String?,

@ColumnInfo(name = "email") val email: String?,

@ColumnInfo(name = "password") val password: String?,

)

UserDao.kt

Package com.example.owlapplication

Import androidx.room.\*

@Dao

Interface UserDao {

    @Query("SELECT \* FROM user\_table WHERE email = :email")

    Suspend fun getUserByEmail(email: String): User?

    @Insert(onConflict = OnConflictStrategy.REPLACE)

    Suspend fun insertUser(user: User)

    @update

    Suspend fun updateUser(user: User)

    @Delete

    Suspend fun deleteUser(user: User)

}

UserDatabase.kt

Package com.example.owlapplication

Import android.content.Context

Import androidx.room.Database

Import androidx.room.Room

Import androidx.room.RoomDatabase

@Database(entities = [User::class], version = 1)

Abstract class UserDatabase : RoomDatabase() {

    Abstract fun userDao(): UserDao



Companion object {

@Volatile

Private var instance: UserDatabase? = null

Fun getDatabase(context: Context): UserDatabase {

Return instance ?: synchronized(this) {

Val newInstance = Room.databaseBuilder(

Context.applicationContext,

UserDatabase::class.java,

“user\_database”

).build()

Instance = newInstance

newInstance

}

}

}

}

UserDatabaseHelper.kt

Package com.example.owlapplication

Import android.annotation.SuppressLint

Import android.content.ContentValues

Import android.content.Context

Import android.database.Cursor

Import android.database.sqlite.SQLiteDatabase

Import android.database.sqlite.SQLiteOpenHelper

Class UserDatabaseHelper(context: Context) :

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

Companion object {

Private const val DATABASE\_VERSION = 1

```

Private const val DATABASE_NAME = "UserDatabase.db"

Private const val TABLE_NAME = "user_table"
Private const val COLUMN_ID = "id"
Private const val COLUMN_FIRST_NAME = "first_name"
Private const val COLUMN_LAST_NAME = "last_name"
Private const val COLUMN_EMAIL = "email"
Private const val COLUMN_PASSWORD = "password"
}

Override fun onCreate(db: SQLiteDatabase?) {
    Val createTable = "CREATE TABLE $TABLE_NAME (" +
        "$COLUMN_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
        "$COLUMN_FIRST_NAME TEXT, " +
        "$COLUMN_LAST_NAME TEXT, " +
        "$COLUMN_EMAIL TEXT, " +
        "$COLUMN_PASSWORD TEXT" +
        ")"

    Db?.execSQL(createTable)
}

Override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int)
{
    Db?.execSQL("DROP TABLE IF EXISTS $TABLE_NAME")
    onCreate(db)
}

Fun insertUser(user: User) {
    Val db = writableDatabase
    Val values = ContentValues()
    Values.put(COLUMN_FIRST_NAME, user.firstName)
    Values.put(COLUMN_LAST_NAME, user.lastName)
    Values.put(COLUMN_EMAIL, user.email)
    Values.put(COLUMN_PASSWORD, user.password)
}

```

```

        Db.insert(TABLE_NAME, null, values)
        Db.close()
    }

    @SuppressWarnings("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
    }

    @SuppressWarnings("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
    }

}

cursor.moveToNext()

    }
    Cursor.close()
    Db.close()
    Return users
}

}
```