

# Owl-M: A MATERIAL DESIGN STUDY APP

Team code : NM2024TMID05452

Name :Ashwinthraj.S.G

Nm id : 02CBEBDA82E3CAAA4A349E26BA3B5A4B

Name :Harivarma .R

Nm id :C9407A4733F532F9198445681D1EDBE9

Name :Devapriyan.G.H

Nm id. :D88F101D30DF87FA2684E1B348BE0B7E

Name :Herena Konecia. P

Nm id :6862E63F92AADFAA72D462BDB47209B1

# Description

- ▶ Study material app: Planning and Research
- ▶
- ▶ Target Audience: Determine who will use your app (students, professionals, etc.).
- ▶
- ▶ Features: List out the features you want in your app (e.g., flashcards, quizzes, progress tracking, etc.).
- ▶
- ▶ Content: Decide on the type of content (text, images, videos) and sources (user-generated, educational institutions).


# Design

1. **Ui & ux**
2. User Interface (UI): Design a user-friendly and intuitive interface. Focus on ease of navigation and accessibility.
- 3.
3. User Experience (UX): Ensure a smooth and engaging user experience. Consider adding gamification elements to keep users motivated.

# Main Activity

```
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,  
    onChange = { 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)  
}
```



```
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.stringReso
```



```
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,
```

```
            )}}}
```




```
Package com.example.owlapplication

import android.os.Bundle

import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent

import androidx.compose.foundation.Imageimport androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollStateimport 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(  

```

```
        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
)
```

```
}
```

```
}
```



```
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
)
```





```
Package com.example.owlapplication

import android.os.Bundle

import androidx.activity.ComponentActivity
import androidx.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 : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        setContentView {

            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(40.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 = Text.Align.Justify,
        fontSize = 16.sp
    ) }

    textAlign = Text.Align.Justify,
    fontSize = 16.sp
}package com.example.owlapplication

import android.os.Bundle

import androidx.activity.ComponentActivity
import androidx.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
```

```
)package com.example.owlapplication
```

```
import android.content.Context
```

```
import android.content.Intent
```

```
import android.os.Bundle
```

```
import androidx.activity.ComponentActivity
```

```
import androidx.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)
```

```
)
```

```

        } 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)

}

```



```
TextField{
```

```
    value = password,
```

```
    onChange = { 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
```



```
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?,  
    )
```



```
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)
```

```
}
```



```
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

            }

        }

    }

}
```



```
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

}
```



```
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)

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)

)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
    )
}
```



```
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

    )

    */
)
```



## Register

Register

Have an account? [Log in](#)



## Login

Login

[Register](#)

[Forget password?](#)

## Study Material



Arts & Craft

The Basics of Woodturning



Painting

An introduction to oil painting



Architecture

City Phenomenon between Urban Structure and Composition



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 the first recorded instances of spindle turning. These illustrations showed a strap a helper used to rotate the lathe while another worker cut the wood.