## OWI-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.

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.Alignmentg

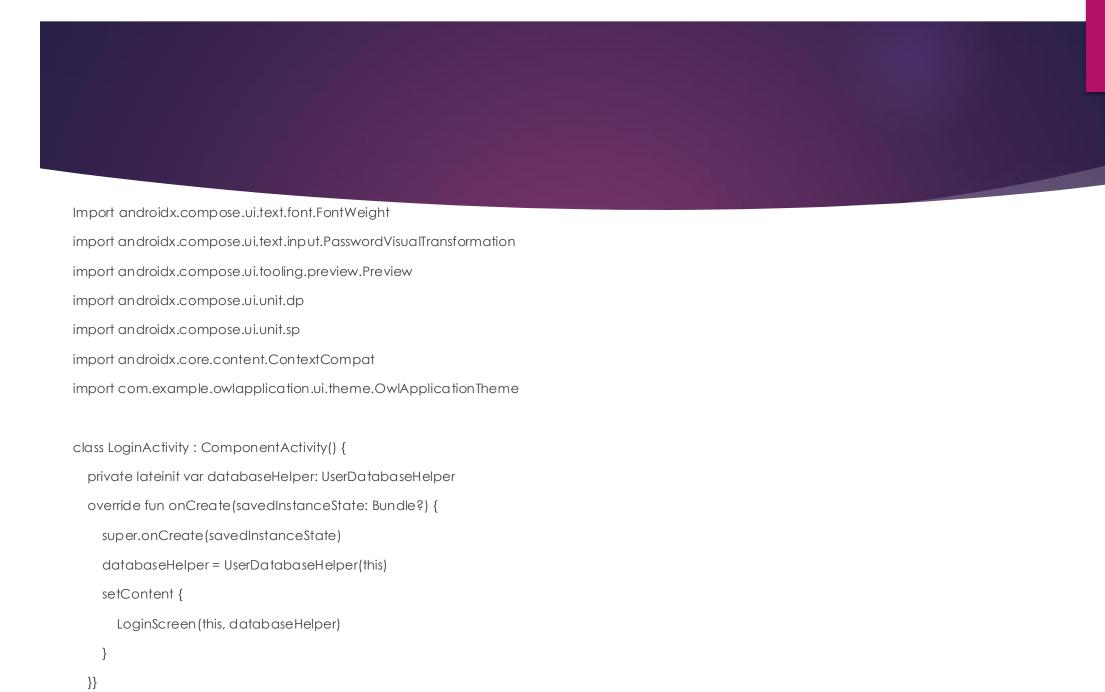
import androidx.compose.ui.Modifier

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



```
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)

color = MaterialTheme.colors.error,

modifier = Modifier.padding(vertical = 16.dp) ) }

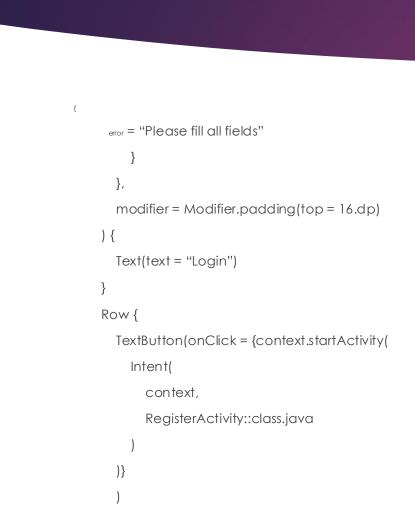
if (error.isNotEmpty()) {

.width(280.dp)

text = error,

Text(

```
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.jav )
             //onLoginSuccess()
           else {
             error = "Invalid username or password"
          } } else
```



```
{ 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.lmage

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())



```
fontSize = 36.sp,
      fontWeight = FontWeight.Bold,
      color = Color(0xFFFF A500),
      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.d ) {
      Column(
        horizontalAlignment = Alignment.CenterHorizontally
      ) {
        Image(
          painterResource(id = R.drawable.img_1), contentDescription = "",
          modifier = Modifier
            .height(150.dp)
             .scale(scale X = 1.2F, scale Y = 1F)
```



```
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(scale X = 1.4F, scale Y = 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
```



### horizontalAlignment = Alignment.CenterHorizontally ) { Image( painterResource(id = R.drawable.img\_3), contentDescription = "", modifier = Modifier .height(150.dp) .scale(scale X = 1.2F, scale Y = 1F) Text (text = string Resource(id = R.string.course3), color = Color(0xFFFF A500), fontSize = 16.sp) Text ( text = stringResource (id = R.string.topic3), fontWeight = FontWeight.Bold, fontSize = 20.sp, textAlign = Text Align.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
      ) {
         lmage(
            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.s 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

 $import\ and roidx. compose. found attion. Image import\ and roidx. compose. found attion. background$ 

import and roidx.compose.foundation.layout.\*

 $import\ and roidx. compose. foundation. remember ScrollState import\ and roidx. compose. foundation. vertical Scroll\ and roidx. compose. foundation.$ 

import androidx.compose.material.Text

import androidx.compose.runtime.Composable

import androidx.compose.ui.Alignment

 $import\ and roidx. compose. ui. Modifier$ 

import androidx.compose.ui.draw.scale import androidx.compose.ui.graphics.Color

import androidx.compose.ui.res.painterResource

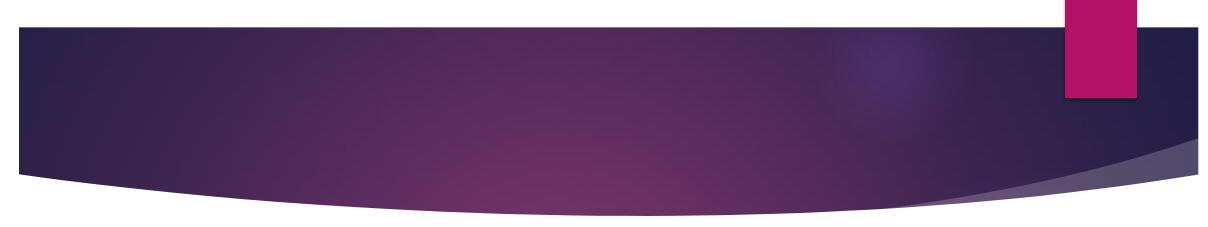
import androidx.compose.ui.res.stringResource

 $import\ and roidx. compose. ui. text. font. Font Weigh\ timport\ and roidx. compose. ui. text. style. Text Align$ 

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

 $import\ com. example. ow lapplication. ui. the me. Ow lApplication Theme$ 



```
dass 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(
```



$$\label{eq:text} \begin{split} \text{text} = & \text{string.Resource (id} = \text{R.string.subheading 1\_1),} \\ \text{mod fier} = & \text{Modifier.align(Alignment.Start),} \end{split}$$

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

$$\label{eq:text} \begin{split} \text{text} = & \text{string,text1\_1} \, , \\ \text{modifier} = & \text{Modifier,align} (\text{Alignment.Start}), \end{split}$$

textAlign = Text Align.Justify,

fontSize = 16.sp

fontSize = 20.sp

Text (



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

Text (

text = stringResource (id = R.string.subhe ading 1_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
)
```



import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll

import androidx.compose.material.Text
import androidx.compose.uui.Alignment
import androidx.compose.ui.Alignment
import androidx.compose.ui.daw.scale
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.fontFontWeight
import androidx.compose.ui.text.fontFontWeight
import androidx.compose.ui.text.style.TextAlign

import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
class MainActivity3 : ComponentActivity() {

override fun on Create (saved Instance State: Bundle?) {

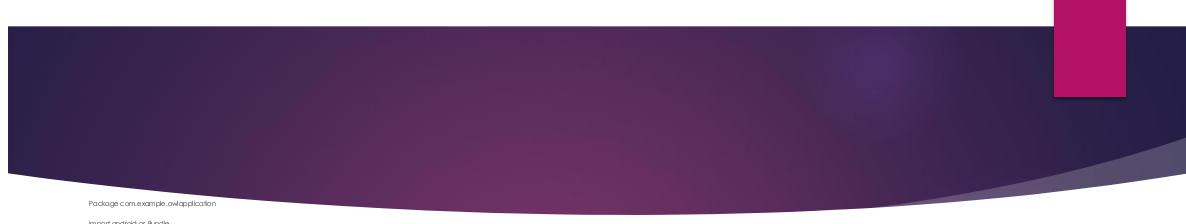
super.onCreate (savedInstanceState)
setContent { Greeting 1()}}}



```
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),
    content Description = "",
    modifier = Modifier.align(Alignment.CenterHorizontally)
      .scale(scale X = 1.2F, scale Y = 1F)
  ) Spacer (modifier = Modifier.height (20.dp)) Text (
   text = stringResource (id = R.string.course2),
   color = Color(0xFFFF A500),
    fontSize = 16.sp,
   modifier = Modifier.align(Alignment.CenterHorizontally) )
  Spacer(modifier = Modifier.height(20.dp))
    text = stringResource (id = R.string.topic2),
   fontWeight = FontWeight.Bold,
    fontSize = 26.sp,
   modifier = Modifier.align(Alignment.CenterHorizontally)
```



```
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 = stringResource (i d = 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 = Text Align.Justify,
 fontSize = 16.sp
```



import android.os.Bundle

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.lmage

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



.verticalSaro II(rememberScrollState (j))
.background (Color.White),
verticalArrangement = Arrangement.Top

painterResource(id = R.drawable.img\_3),

.scale(scale X = 1.5F, scale Y = 2F)) Spacer(modifier = Modifier.height(60.dp))

text = stringResource (id = R.string.course3),

modifier = Modifier.align(Alignment.CenterHorizontally)

modifier = Modifier.align(Alignment.CenterHorizontally)

content Description = "",

color = Color(0xFFFF A500), fontSize = 16.sp,

) { | Image(

Text (



modifier = Modifier.align(Alignment.Start),

 $\label{eq:spacer} Spacer\{modifier = Modifier.height(20.dp)\} \ \ \, Text\{ \\ text = stringResource \{id = R.string.text3\_1\}, \\ modifier = Modifier.align(Alignment.Start), \\ \end{cases}$ 

fontSize = 16.sp ) Spacer(modifier = Modifier.height(20.dp))

$$\label{eq:continuity} \begin{split} \text{text} = & \text{string.subhe ading 3\_2} \text{)}, \\ \text{mod fier} = & \text{Mod fier.align(Alignment.Start)}, \end{split}$$

fontSize = 20.sp }

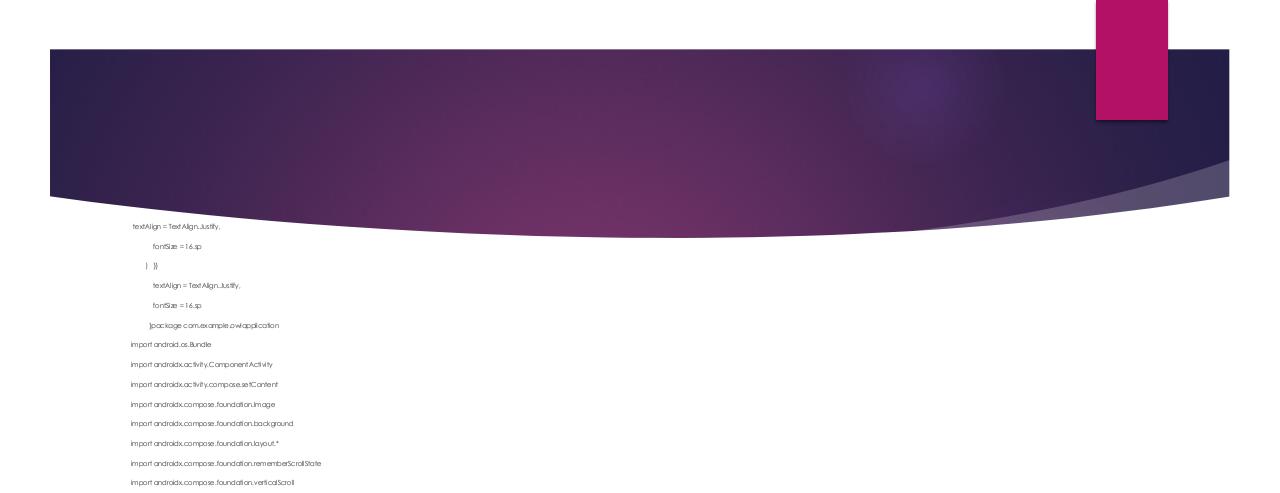
textAlign = TextAlign.Justify,

fontSize = 20.sp )

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

text = stringResource (id = R.string.text3\_2),
modifier = Modifier.align(Alignment.Start),

Text (



import androidx.compose.material.MaterialTheme import androidx.compose.material.Surf ace 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.graphics.Color import androidx.compose.ui.res.painterResource import androidx.compose.ui.res.stringResource import androidx.compose.ui.res.stringResource import androidx.compose.ui.res.stringResource import androidx.compose.ui.text.fontFontWeight



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), content Description = "", modifier = Modifier.align(Alignment.CenterHorizontally) .scale(scaleX = 1.5F, scaleY = 1.5F)) Spacer(modifier = Modifier.height(60.dp)) Text (



text = stringResource (id = R.string.topic4),
fontWeight = FontWeight.Bold,

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

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

$$\label{eq:text} \begin{split} \text{text} = & \text{string.Resource (id} = \text{R.string.text4\_1}), \\ \text{modifier} = & \text{Modifier.align(Alignment.Start)}, \end{split}$$

 $\label{eq:spacer} Spacer(modifier = Modifier.height(20.dp Text($$ text = string.Resource (id = R.string.subhe.ading4_2)$, $$ modifier = Modifier.align(Alignment.Start)$,$ 

$$\label{eq:text} \begin{split} \text{text} = & \text{string.Resource (id} = \text{R.string.subheading4\_1),} \\ \text{modifier} = & \text{Modifier.align(Alignment.Start),} \end{split}$$

modifier = Modifier.align(Alignment.CenterHorizontally) )

fontSize = 26.sp,

fontSize = 20.sp )

textAlign = Text Align.Justify, fontSize = 16.s )

fontSize = 20.sp )

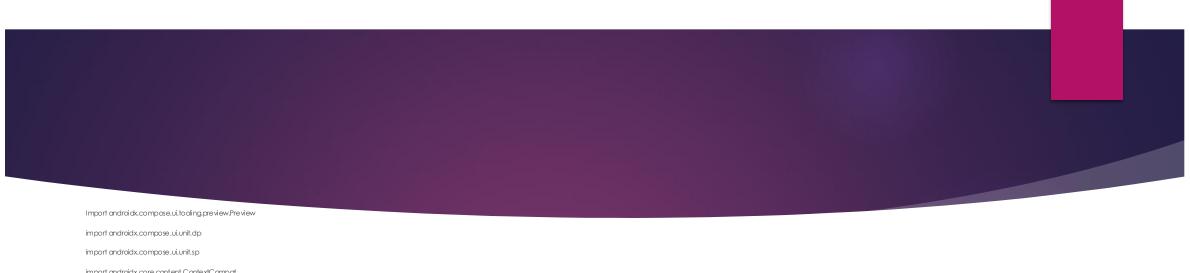
Text (

Text (



modifier = Modifier.align(Alignment.Start), textAlign = Text Align.Justify, fontSize = 16.sp ) package com.example.owlapplication import android.content.Context import android.content.Intent import android.os.Bundle import androidx.activity.Component Activity import androidx.activity.compose.setContent import androidx.compose.foundation.lmage 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.fontFontWeight

import androidx.compose.ui.text.input.PasswordVisualTransformation



import androidx.core.cont ent.ContextCompat import com.example.owlapplication.ui.theme.OwlApplicationTheme class RegisterActivity: ComponentActivity() { private lateinit var databaseHelper: UserDatabaseHelper override fun on Create (saved Instance State: 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



```
fontSize = 36.sp,
 fontWeight = FontWeight.ExtraBold,
  fontFamily = FontFamily.Cursive,
  text = "Register"
Spacer(modifier = Modifier.height(10.dp))
Text Field(
  value = username,
  onValueChange = { username = it },
  label = { Text ("Username") },
  modifier = Modifier
    .padding(10.dp)
    .width(280.dp)
Text Field(
  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) {
  valintent = Intent(context, LoginActivity::class.java)
  ContextCompat.startActivity(context, intent, null)
```



```
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,
        first Name = username,
        lastName = null,
         em ail = email,
        password = password
       databaseHelper.insertUser(user)
       error = "User registered successfully"
       // Start LoginActivity using the current context
```

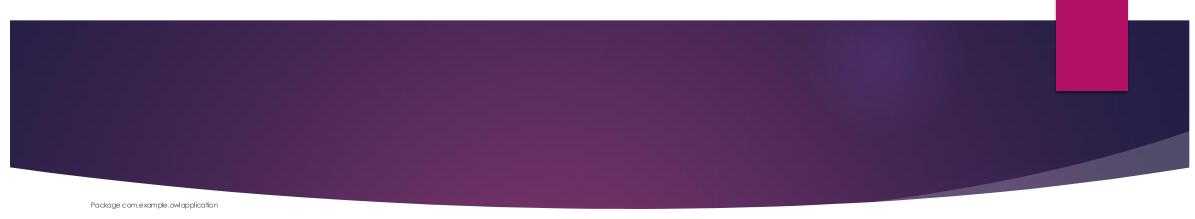


```
import androidx.room.ColumnInfo
import androidx.room.Entity
import androidx.room.PrimaryKey

@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 e mail = :email")
  suspend fun get UserByEmail(email:String): User?
  @Insert (onConflict = OnConflictStrategy.REPLACE)
  suspend fun insertUser(user: User)
  @Update
  suspend fun updateUser(user:User)
  @Delete
  suspend fun delete User (user: User)
```



```
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
```



companion object {

private const val DATABASE\_VERSION = 1

private const val TABLE\_NAME = "user\_table"

private const val COLUMN\_ID = "id"

private const val COLUMN\_EMAIL = "email"

} override fun onCreate(db: SQLiteDatabase?) {
 val createTable = "CREATE TABLE \$TABLE\_NAME ("+

"\$COLUMN\_FIRST\_NAME TEXT, " +

private const val DATABASE\_NAME = "UserDatabase.db"

private const val COLUMN\_FIRST\_NAME = "first\_name"

private const val COLUMN\_LAST\_NAME = "last\_name"

private const val COLUMN\_PASSWORD = "password"

"\$COLUMN\_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +



values.put(COLUMN\_FIRST\_NAME, user.firstName)
values.put(COLUMN\_LAST\_NAME, user.lastName)

values.put(COLUMN\_PASSWORD, user.password)

fun getUserByUsemame(username: String): User? {

val cursor: Cursor = db.rawQuery("SELECT\* FROM \$TABLE\_NAME WHERE \$COLUMN\_FIRST\_NAME = ?", arrayOf(username))

values.put(COLUMN\_EMAIL, user.email)

db.insert(TABLE\_NAME, null, values)

db.close()

}@SuppressLint("Range")

var us er: User? = nul
lif (curs or.move ToFirst()) {
 user = User(

val db = readableDatabase



```
email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
      password = cursor.get String(cursor.getColumnIndex(COLUMN_PASSWORD)),
   ) }
  cursor.close()
  db.close()
  return user }
@SuppressLint ("Range")
fun getUserByld(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)),
      first Name = cursor.get String(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
      lastName = cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
      email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
      password = cursor.get String(cursor.getColumnIndex(COLUMN_PASSWORD)),
  cursor.close()
  db.close()
  return user
```



```
@SuppressLint ("Range")
fun getAllUsers(): List<User> {
  valusers = 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)),
        first Name = cursor.get String(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
```



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 mport 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



)@Composable fun OwlApplicationTheme(

}else{

LightColorPalette

MaterialTheme(

colors = colors,

typography = Typography,
shapes = Shapes,
content = content}}

darkTheme: Boolean = isSystemInDarkTheme(),



```
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 Typo graphy = Typography(
  body 1 = 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
```

10:43 🕅 🖸

%42 ¼ \*fil ■2



### Register

Username

Email

Password

Register

Have an account? Log in



12.0 % \*fail 🐷

Login

Username

10:43 № 🖸

Password

Login

Register

Forget password?

10:44 № 🖸

QQ5 独 指 (室)

### **Study Material**



The Basics of Woodturning



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.