A logo with columns and stars

Description automatically generatedA black and grey logo with a book and a black hat

Description automatically generated with medium confidence**A logo of a building with columns

Description automatically generated**

**ADAPTIVE MAIL: A FLEXIBLE E-MAIL CLIENT APP**

**A PROJECT REPORT**

**SUBMITTED BY**

**Arun. P.K - 812022104010**

**Arul Kumaran A- 812022104008**

**Chandru - 812022104017**

**Kishore - 812022104039**

**BACHELOR OF ENGINEERING**

**IN**

**FIFTH SEMESTER**

**COMPUTER SCIENCE AND ENGINEERING**

**M.A.M. COLLEGE OF ENGINEERING AND TECHNOLOGY, TRICHY**

**ANNA UNIVERSITY : CHENNAI 600 025 NOVEMBER 2024**

## **1. Introduction**

**Owl-M** is an interactive learning application designed to help users understand and implement **Material Design principles** effectively. By combining theoretical knowledge with hands-on practice, the app empowers UI/UX designers, developers, and design enthusiasts to explore modern design standards.

## **2. Objectives**

1. **Educate** users about Material Design concepts like color theory, typography, iconography, motion, and layout.
2. **Provide Tools** for experimenting with Material Design components in a sandbox environment.
3. **Promote Learning** through interactive modules, quizzes, and real-world examples.
4. **Foster Creativity** by enabling users to create, test, and share their design projects

## **3. App Features**

### **3.1 Learning Modules**

* Comprehensive lessons on Material Design principles.
* Interactive demonstrations of key concepts.

### **3.2 Customization Tools**

* A sandbox for experimenting with Material Design themes, layouts, and components.
* Dynamic theming with support for **light** and **dark modes**.

### **3.3 Real-Time Code Integration**

* Real-time previews of design changes.
* Code snippets that show how to implement Material Design components.

### **3.4 Gamification**

* Quizzes to test understanding.
* Achievements and badges to motivate users.

### **3.5 Community Sharing**

* Share and discover design projects created by other users.
* Collaborate and provide feedback on shared templates.

## **4. Design Philosophy**

### **4.1 Material Design Guidelines**

The app strictly adheres to Google's **Material Design Guidelines**, ensuring:

* **Consistent UI Elements**: Buttons, cards, and other components follow Material Design standards.
* **Accessibility**: Color contrast, font sizes, and layouts cater to diverse user needs.
* **Responsive Design**: Optimized for devices of all sizes.

### **4.2 Visual Aesthetic**

* Clean, intuitive interfaces.
* Use of animations for seamless transitions and enhanced user experience.

## **5. Technical Overview**

### **5.1 Platform and Framework**

* **Developed Using**: Flutter (cross-platform framework).
* **Programming Language**: Dart.
* **Deployment Platforms**: Android, iOS, and Web.

### **5.2 Key Technologies**

* **Google Fonts**: For typography customizations.
* **Provider**: For state management.
* **Material Design Components**: To build and showcase UI elements.

### **5.3 Architecture**

* **Layered Architecture**:
  + **UI Layer**: Screens and widgets.
  + **Business Logic Layer**: Handles learning modules, quizzes, and customization logic.
  + **Data Layer**: Stores user progress and preferences.

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(0xFFFFA500),

modifier = Modifier.align(Alignment.CenterHorizontally))

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

// 01

Card(

modifier = Modifier

.fillMaxWidth()

.height(250.dp)

.clickable {

context.startActivity(

Intent(context, MainActivity2::class.java)

)

},

elevation = 8.dp

)

{

Column(

horizontalAlignment = Alignment.CenterHorizontally

) {

Image(

painterResource(id = R.drawable.img\_1), contentDescription = "",

modifier = Modifier

.height(150.dp)

.scale(scaleX = 1.2F, scaleY = 1F)

)

Text(text = stringResource(id = R.string.course1),color = Color(0xFFFFA500),

fontSize = 16.sp)

Text(

text = stringResource(id = R.string.topic1),

fontWeight = FontWeight.Bold,

fontSize = 20.sp,

textAlign = TextAlign.Center,

)

}

}

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

// 02

Card(

modifier = Modifier

.fillMaxWidth()

.height(250.dp)

.clickable {

context.startActivity(

Intent(context, MainActivity3::class.java)

)

},

elevation = 8.dp

)

{

Column(

horizontalAlignment = Alignment.CenterHorizontally

) {

Image(

painterResource(id = R.drawable.img\_2), contentDescription = "",

modifier = Modifier

.height(150.dp)

.scale(scaleX = 1.4F, scaleY = 1F)

)

Text(text = stringResource(id = R.string.course2),color = Color(0xFFFFA500),

fontSize = 16.sp)

Text(

text = stringResource(id = R.string.topic2),

fontWeight = FontWeight.Bold,

fontSize = 20.sp,

textAlign = TextAlign.Center,

)

}

}

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

// 03

Card(

modifier = Modifier

.fillMaxWidth()

.height(250.dp)

.clickable {

context.startActivity(

Intent(context, MainActivity4::class.java)

)

},

elevation = 8.dp

)

{

Column(

horizontalAlignment = Alignment.CenterHorizontally

) {

Image(

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

modifier = Modifier

.height(150.dp)

.scale(scaleX = 1.2F, scaleY = 1F)

)

Text(text = stringResource(id = R.string.course3),color = Color(0xFFFFA500),

fontSize = 16.sp)

Text(

text = stringResource(id = R.string.topic3),

fontWeight = FontWeight.Bold,

fontSize = 20.sp,

textAlign = TextAlign.Center,

)

}

}

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

// 04

Card(

modifier = Modifier

.fillMaxWidth()

.height(250.dp)

.clickable {

context.startActivity(

Intent(context, MainActivity5::class.java)

)

},

elevation = 8.dp

)

{

Column(

horizontalAlignment = Alignment.CenterHorizontally

) {

Image(

painterResource(id = R.drawable.img\_4), contentDescription = "",

modifier = Modifier

.height(150.dp)

.scale(scaleX = 1.2F, scaleY = 1F)

)

Text(text = stringResource(id = R.string.course4),color = Color(0xFFFFA500),

fontSize = 16.sp)

Text(

text = stringResource(id = R.string.topic4),

fontWeight = FontWeight.Bold,

fontSize = 20.sp,

textAlign = TextAlign.Center,

)

}

}

}

}

Mainactivity2.Kt:

package com.example.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

)

}

}

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()

}

@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())

}

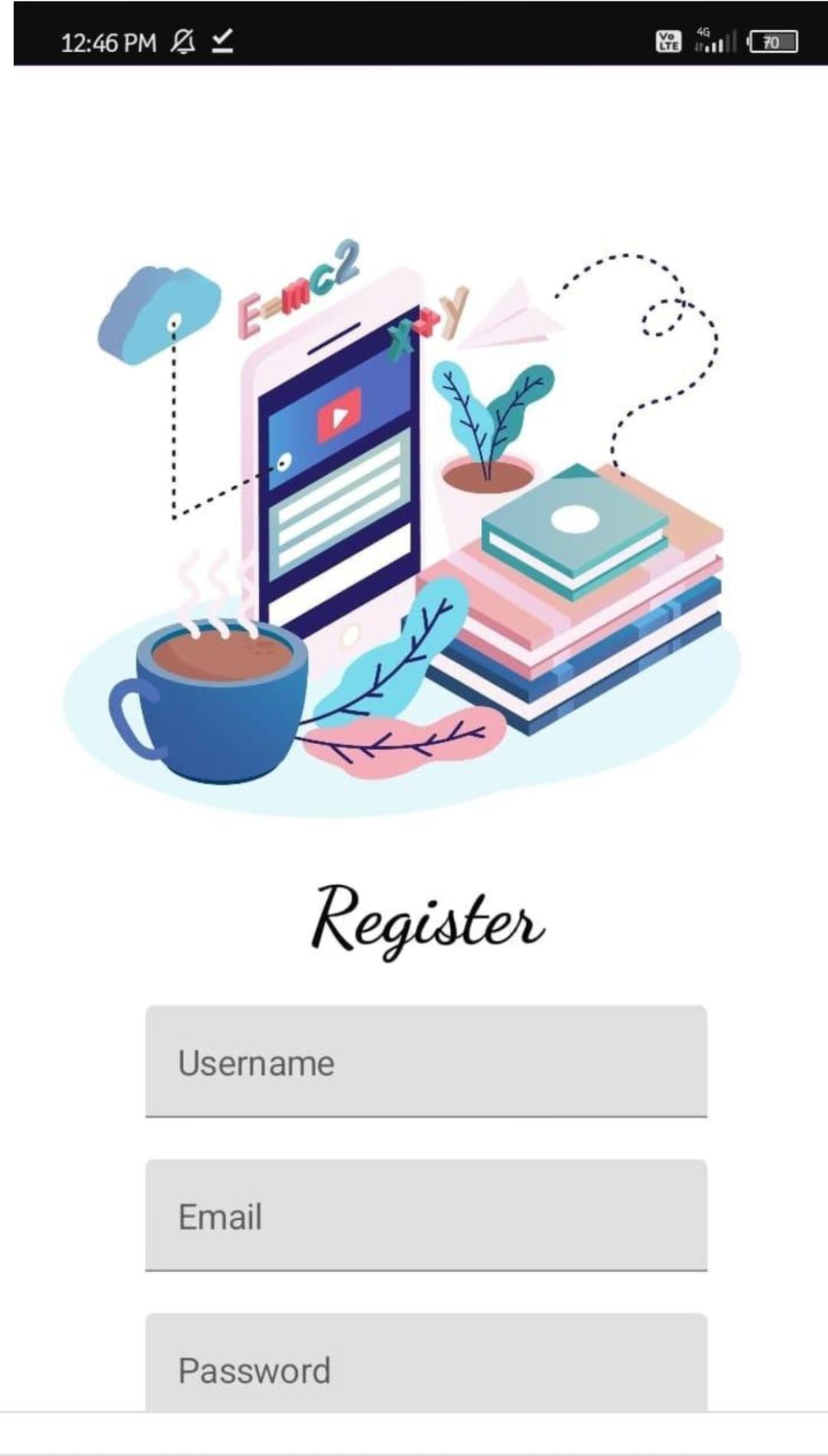
cursor.close()

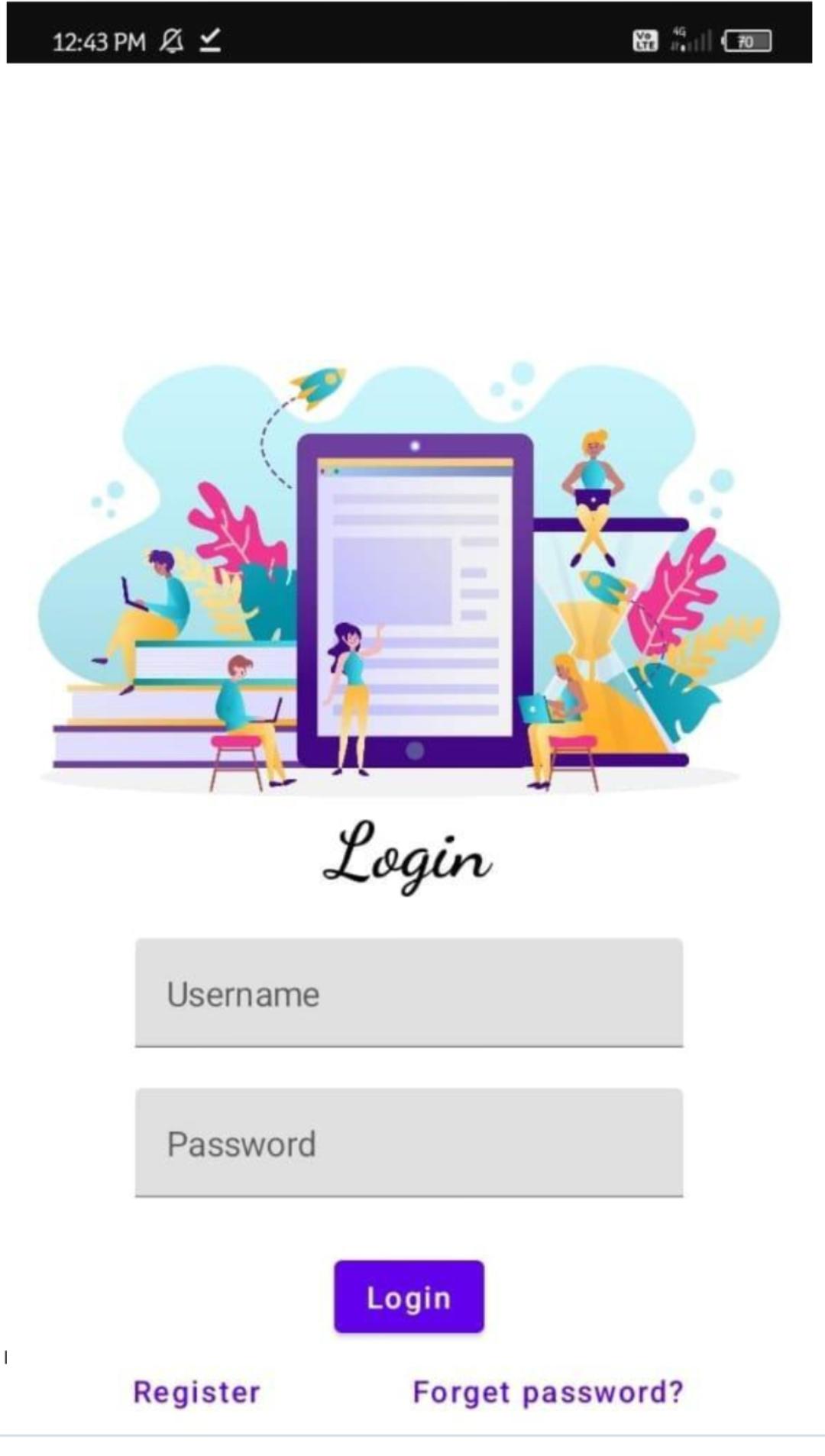
db.close()

return users

}

}

**Output**



## **6. Benefits**

### **6.1 For Designers**

* Learn Material Design principles with practical examples.
* Experiment with design ideas in a controlled environment.

### **6.2 For Developers**

* Understand implementation of Material Design through code snippets.
* Save time by previewing and testing designs before development.

### **6.3 For Teams and Organizations**

* Train members on design consistency and standards.
* Foster collaboration on shared design templates

## **7. Future Roadmap**

### **7.1 Advanced Features**

* Expand content to cover **AR/VR design** and **inclusive design principles**.

### Add **collaborative design tools** for team p**projects.7.2 Platform Enhancements**

* Offline mode for learning on-the-go.
* Integration with design tools (e.g., Sketch, Figma).

### **7.3 Community Growth**

* Build a larger user base through social media and online workshops.
* Host design challenges and hackathons

## **Conclusion**

Owl-M serves as a comprehensive tool for learning and applying Material Design principles. Its interactive approach, robust features, and commitment to accessibility make it a valuable resource for both beginners and experienced professionals