A Material Design Study App

1. INTRODUCTION

1.1Overview

A material design study app developed in Android Studio is a mobile application designed to provide a modern and user-friendly learning experience for students. The app utilizes Google's Material Design principles to create a visually appealing and intuitive interface that makes it easy for users to navigate and interact with educational content. The app provides interactive and engaging learning experiences, with features such as quizzes, flashcards, and games. This helps to make learning more fun and enjoyable, while also improving retention and comprehension.

The app can be customized to suit the needs and preferences of individual users. For example, users can set learning goals, track their progress, and receive personalized recommendations based on their performance. The app is designed to be accessible to users with different abilities and needs. It includes features such as text-to-speech, adjustable font sizes, and high contrast mode.

The app can be integrated with learning management systems (LMS) to provide a seamless learning experience. This enables users to access educational resources, track their progress, and communicate with teachers and peers through a single platform. The app can provide users with access to a community of learners and educators, allowing them to collaborate, share knowledge, and receive feedback.

1.2 Purpose

A Material Design study app can help users learn about various aspects of Material Design, including typography, color theory, layout guidelines. By using the app, users can improve their design skills and create better-looking and more initiative apps that follow the Material Design guidelines.

The app can serve as a reference guide for users to access information on material design principles and best practices while developing Android apps.

Material Design Study App could help users to understand the various components that make up Material Design, such as buttons, cards, and navigation drawers.

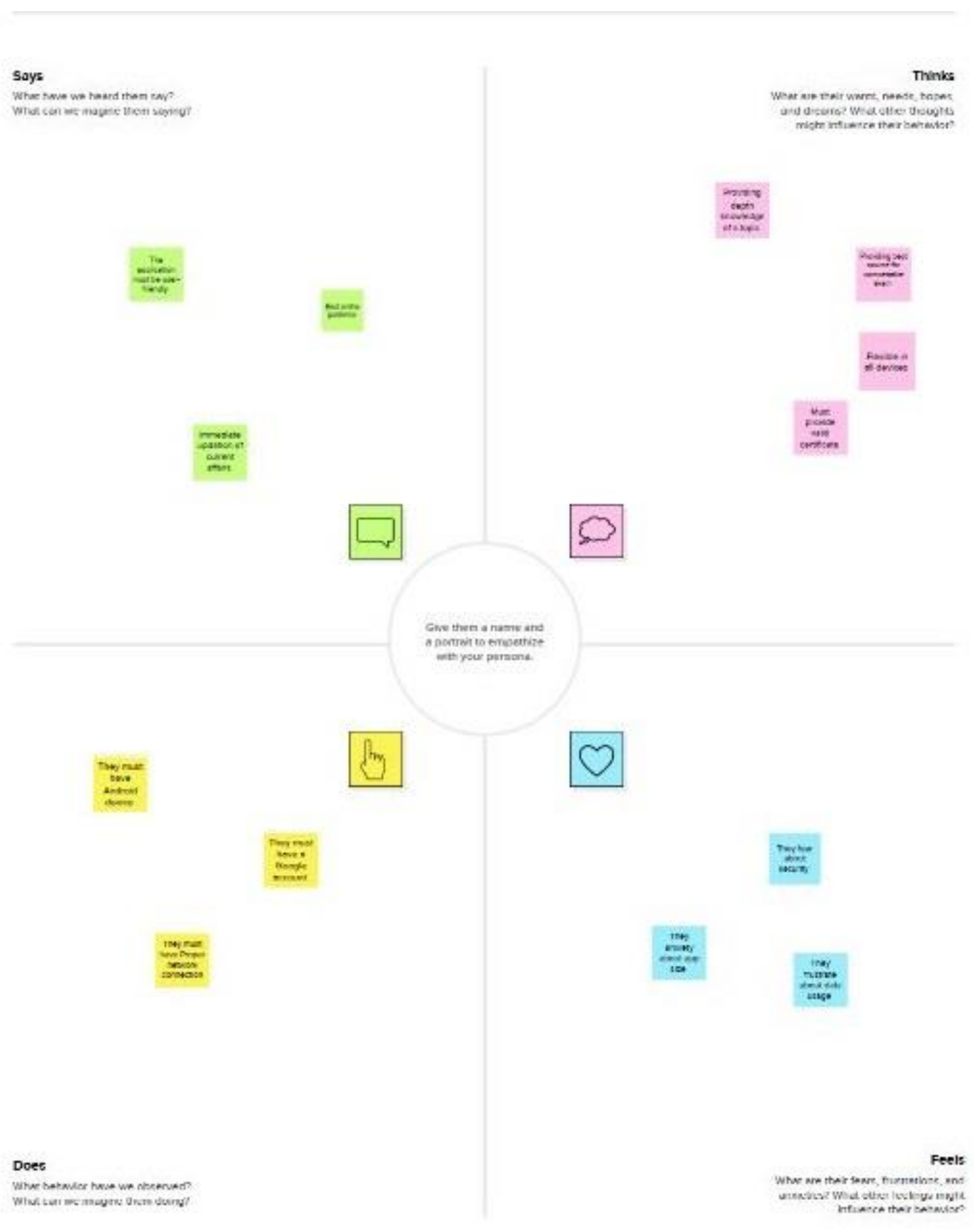
Users could learn how to use these components in their own designs to create consistent user experiences.

The app could provide a library of Material Design examples, allowing users to explore how different components are used in real-world designs.

Overall, the purpose of a Material Design Study App would be to provide users with a comprehensive platform to learn, practice, and apply Material Design principles in their own designs.

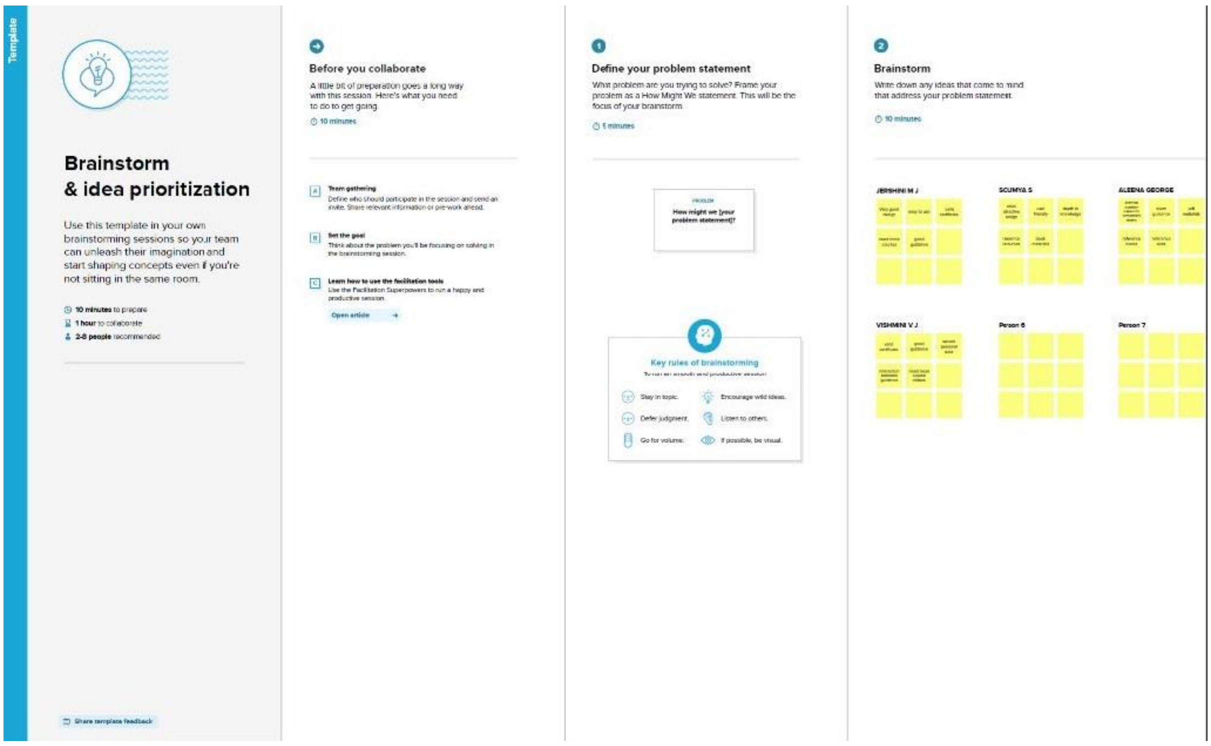
2.Problem Definition & Design Thinking

2.1 Empathy Map



Says
What have we heard them say?
What can we imagine them saying?
The application must be user-friendly
Immediate updation of current affairs
Thinks
What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?
Providing depth knowledge of a topic
Flexible in all devices
Must provide valid certificate
Give them a name and a portrait to empathize with your persona.
They must have Android device
They must have a Google account
They must have Proper network connection
Does
What behavior have we observed?
What can we imagine them doing?
They fear about security
They anxiety about app size
They frustrate about data usage
Feels
What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

2.2 Ideation and Brainstorming Map



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering** Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal** Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools** Use the Facilitation Superpowers to run a happy and productive session.

Open article

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM
How might we [your problem statement]?

Key rules of brainstorming
To run an smooth and productive session

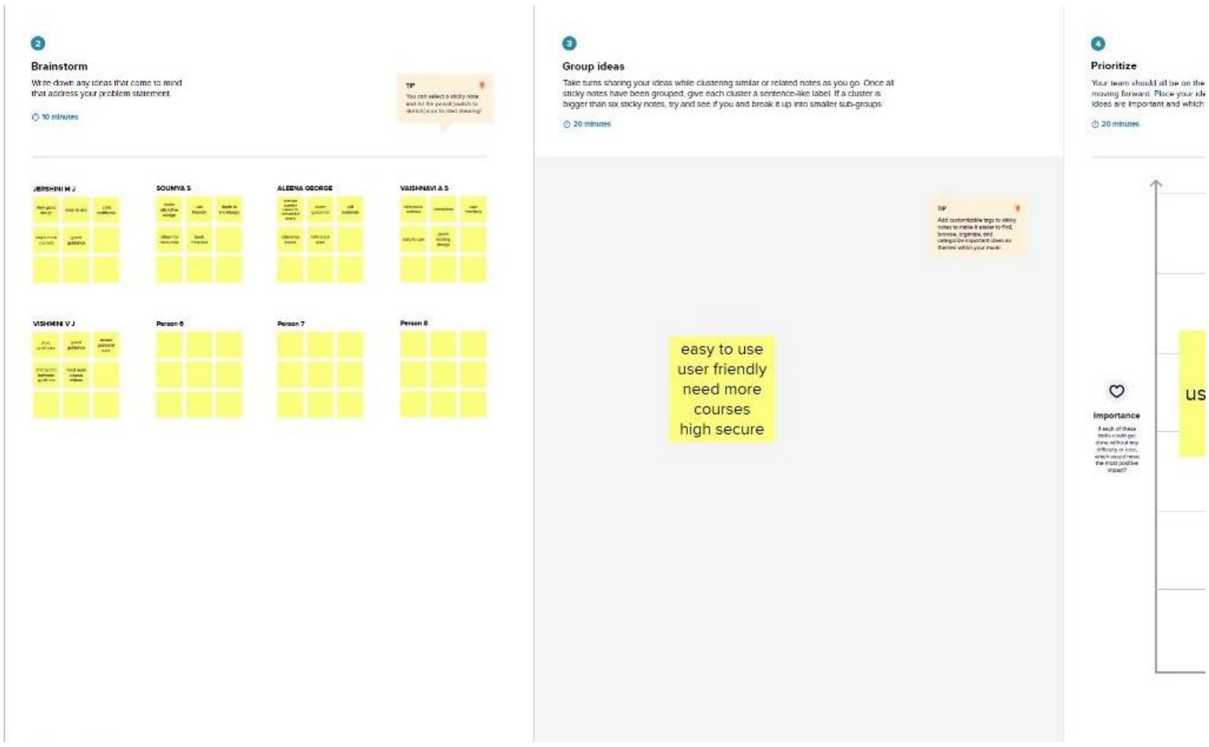
Stay in topic. Encourage wild ideas.
Defer judgment. Listen to others.
Go for volume. If possible, be visual.

2 **Brainstorm**

Write down any ideas that come to mind that address your problem statement.

10 minutes

JERSHINI M J
SOUMYA S
ALEENA GEORGE
VISHMINI V J
Person 6
Person 7



2 **Brainstorm**

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

JERSHINI M J
SOUMYA S
ALEENA GEORGE
VAISHNAVI A S
VISHMINI V J
Person 6
Person 7
Person 8

3 **Group ideas**

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

easy to use
user friendly
need more
courses
high secure

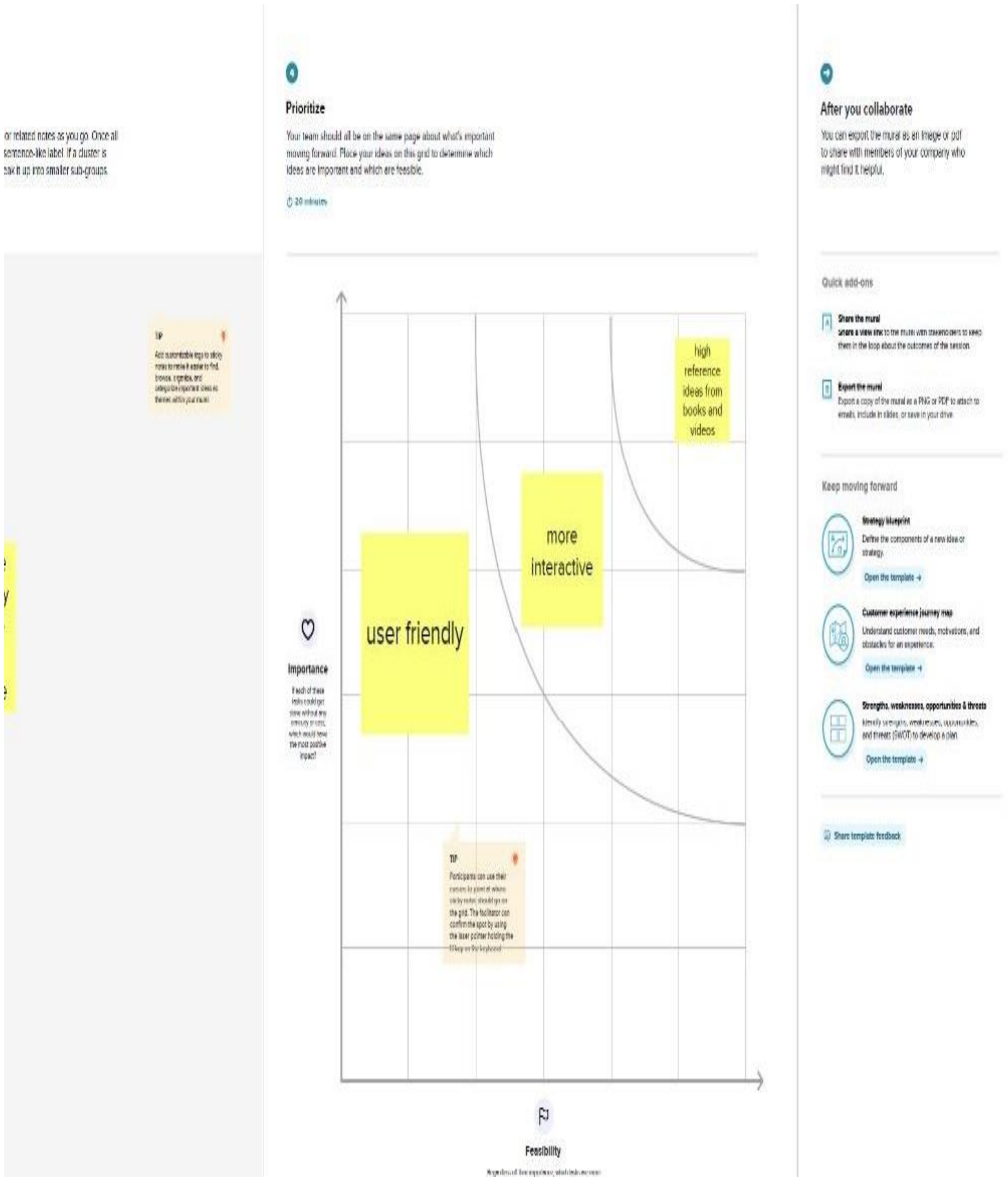
4 **Prioritize**

Your team should all be on the
moving forward. Place your ide
ideas are important and which

20 minutes

Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

us



or related notes as you go. Once all
sentence-like label. If a cluster is
eak it up into smaller sub-groups.

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

high reference ideas from books and videos
more interactive
user friendly
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are easiest

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the [illegible] key on the keyboard.

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

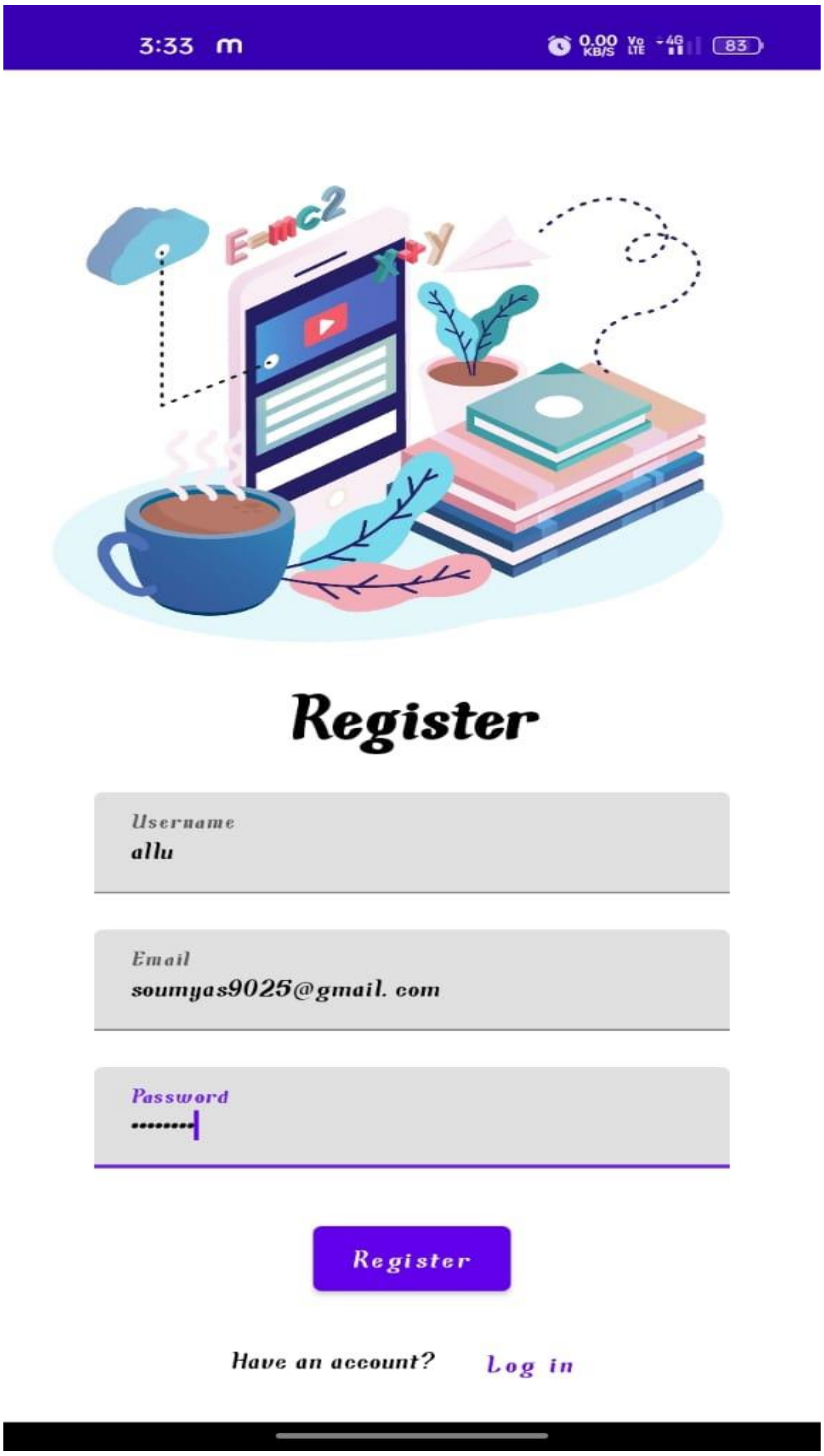
Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

2. RESULT

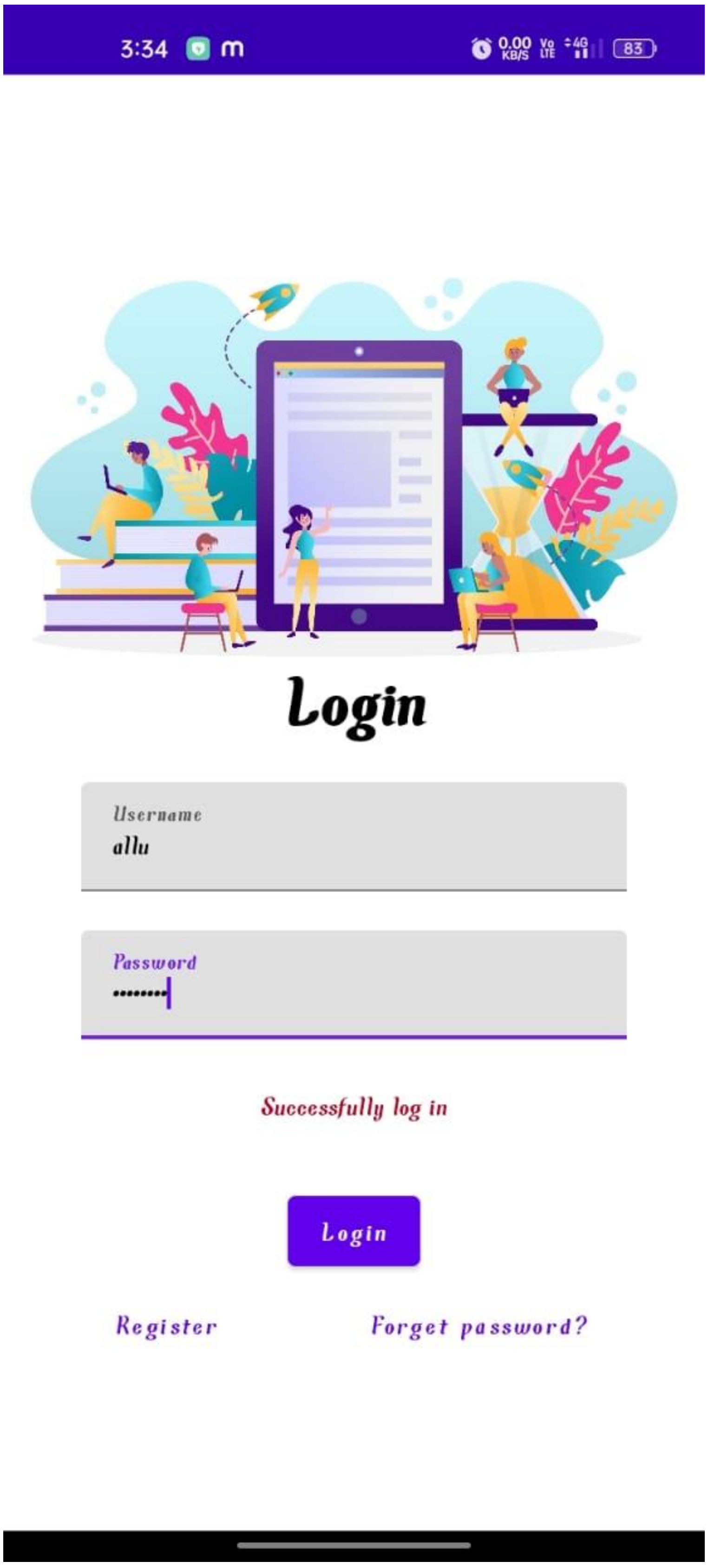

3:33

Register

Username
allu

Email
soumyas9025@gmail. com

Password
••••••••



3:34

Login

Username
allu

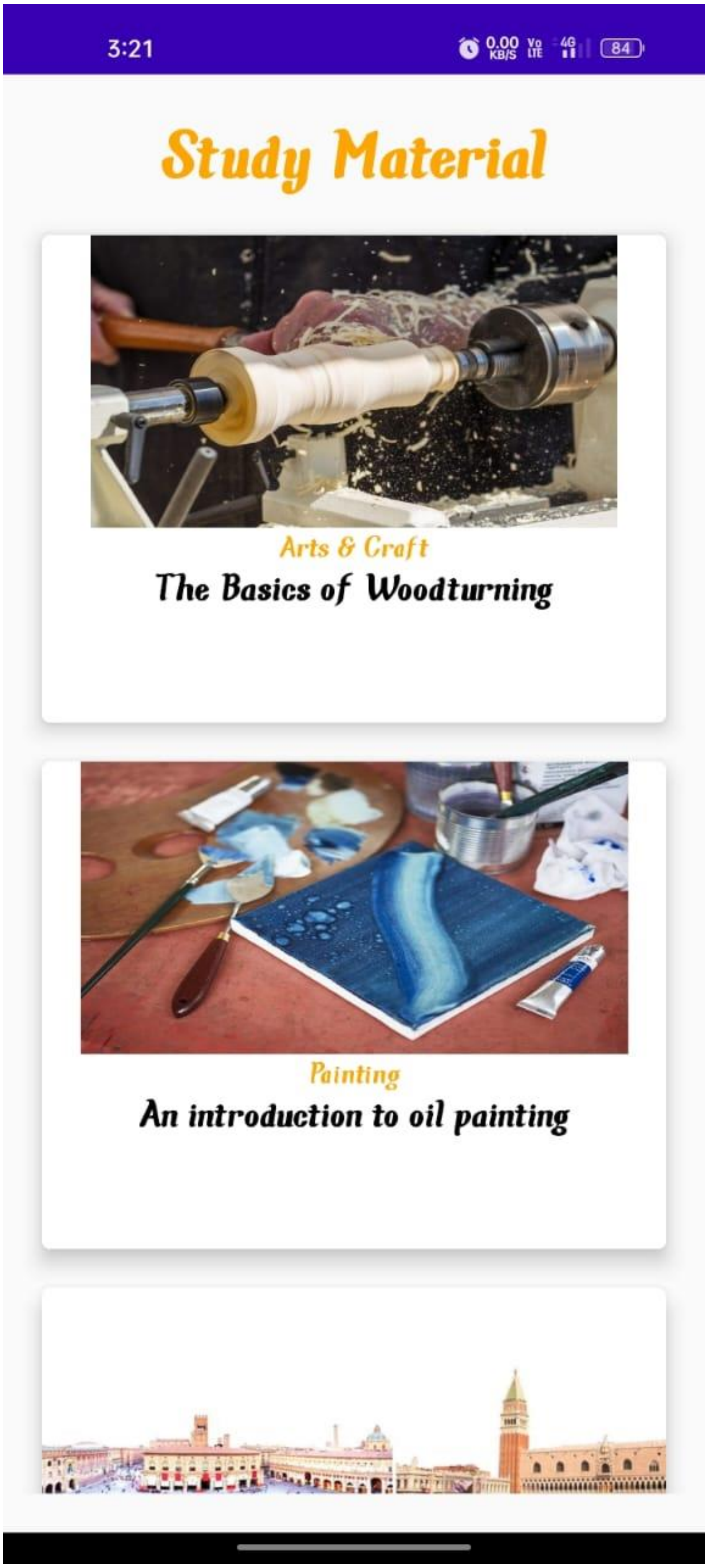
Password
••••••••

Successfully log in

Login

Register

Forget password?



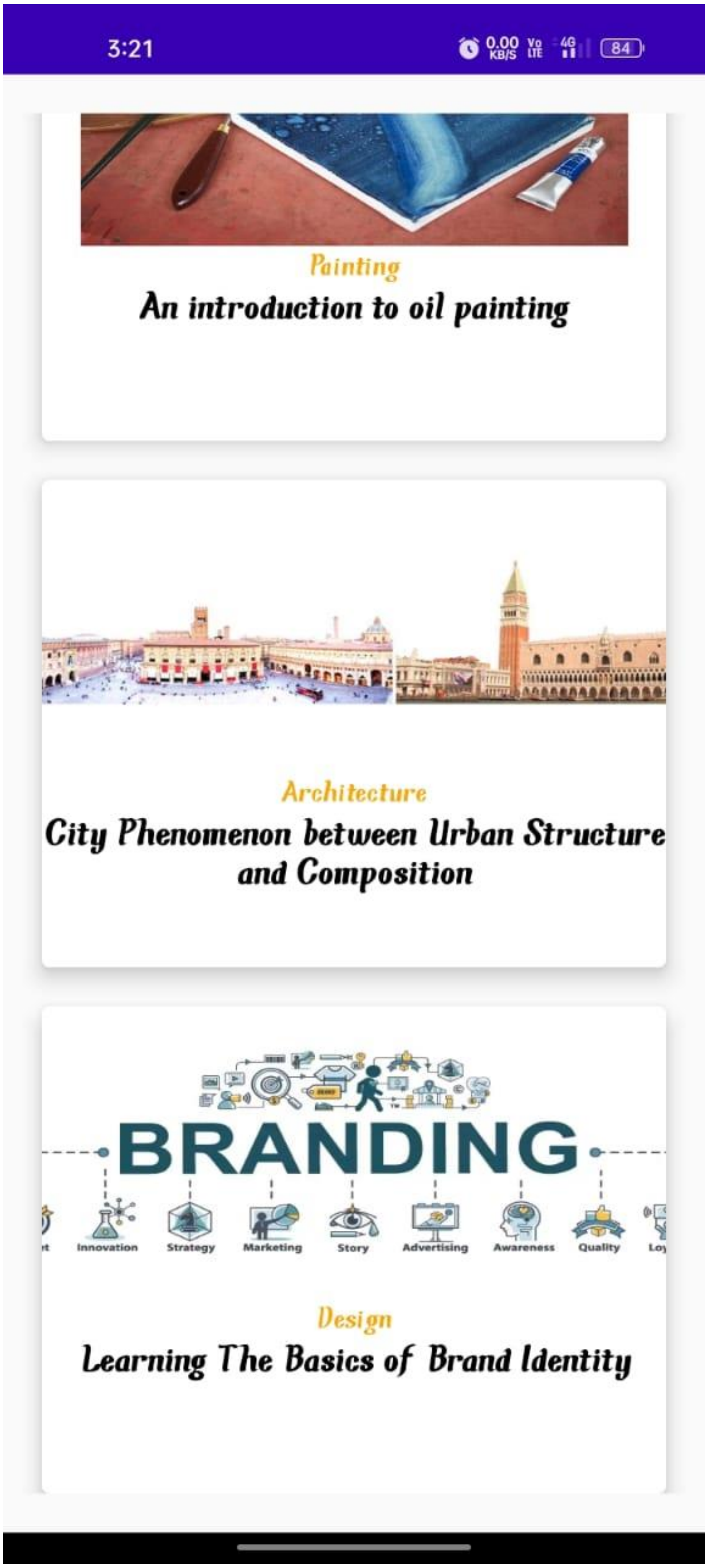
Study Material

Arts & Craft

The Basics of Woodturning

Painting

An introduction to oil painting



3:21

Painting

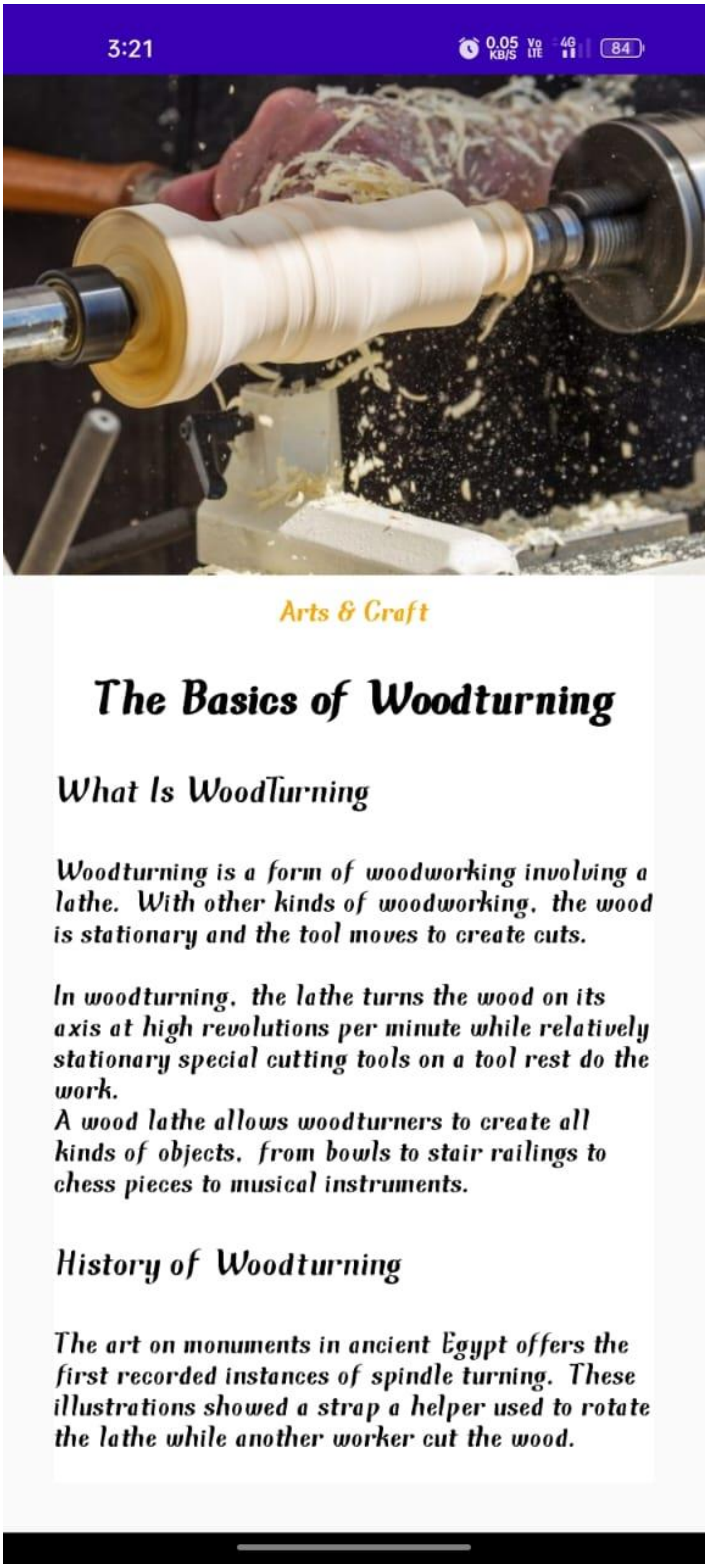
An introduction to oil painting

Architecture

City Phenomenon between Urban Structure and Composition

Design

Learning The Basics of Brand Identity



Arts & Craft

The Basics of Woodturning

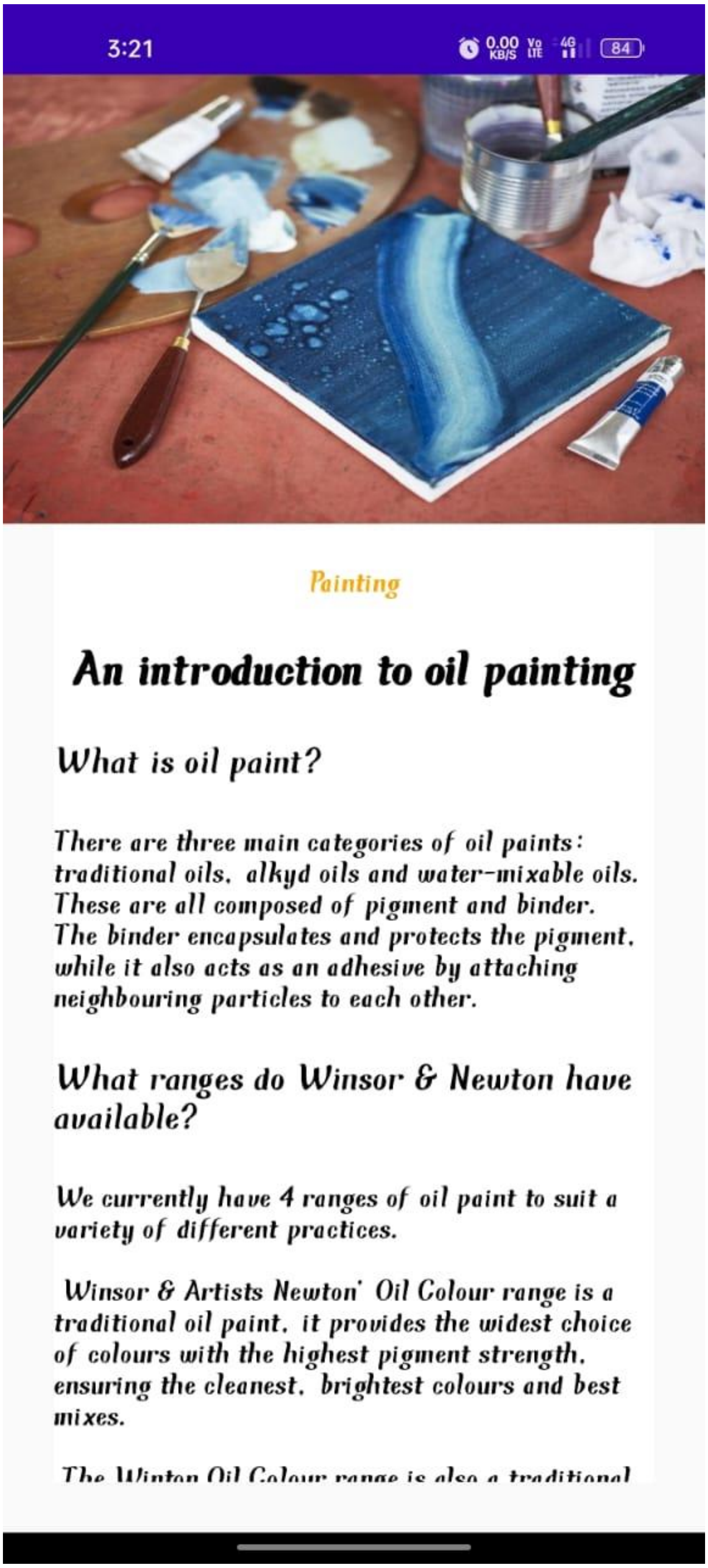
What Is WoodTurning

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.



Painting

An introduction to oil painting

What is oil paint?

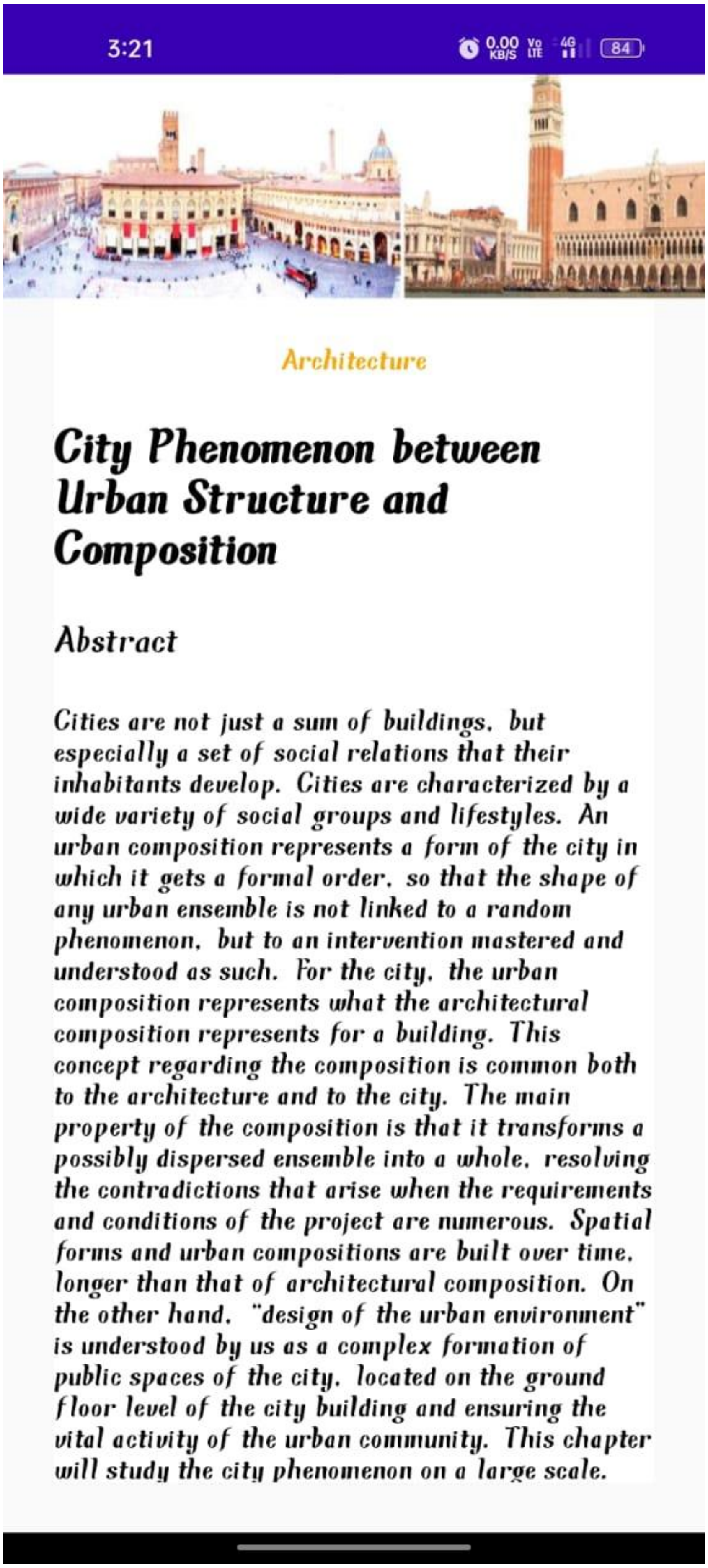
There are three main categories of oil paints: traditional oils, alkyd oils and water-mixable oils. These are all composed of pigment and binder. The binder encapsulates and protects the pigment, while it also acts as an adhesive by attaching neighbouring particles to each other.

What ranges do Winsor & Newton have available?

We currently have 4 ranges of oil paint to suit a variety of different practices.

Winsor & Artists Newton' Oil Colour range is a traditional oil paint, it provides the widest choice of colours with the highest pigment strength, ensuring the cleanest, brightest colours and best mixes.

The Winton Oil Colour range is also a traditional

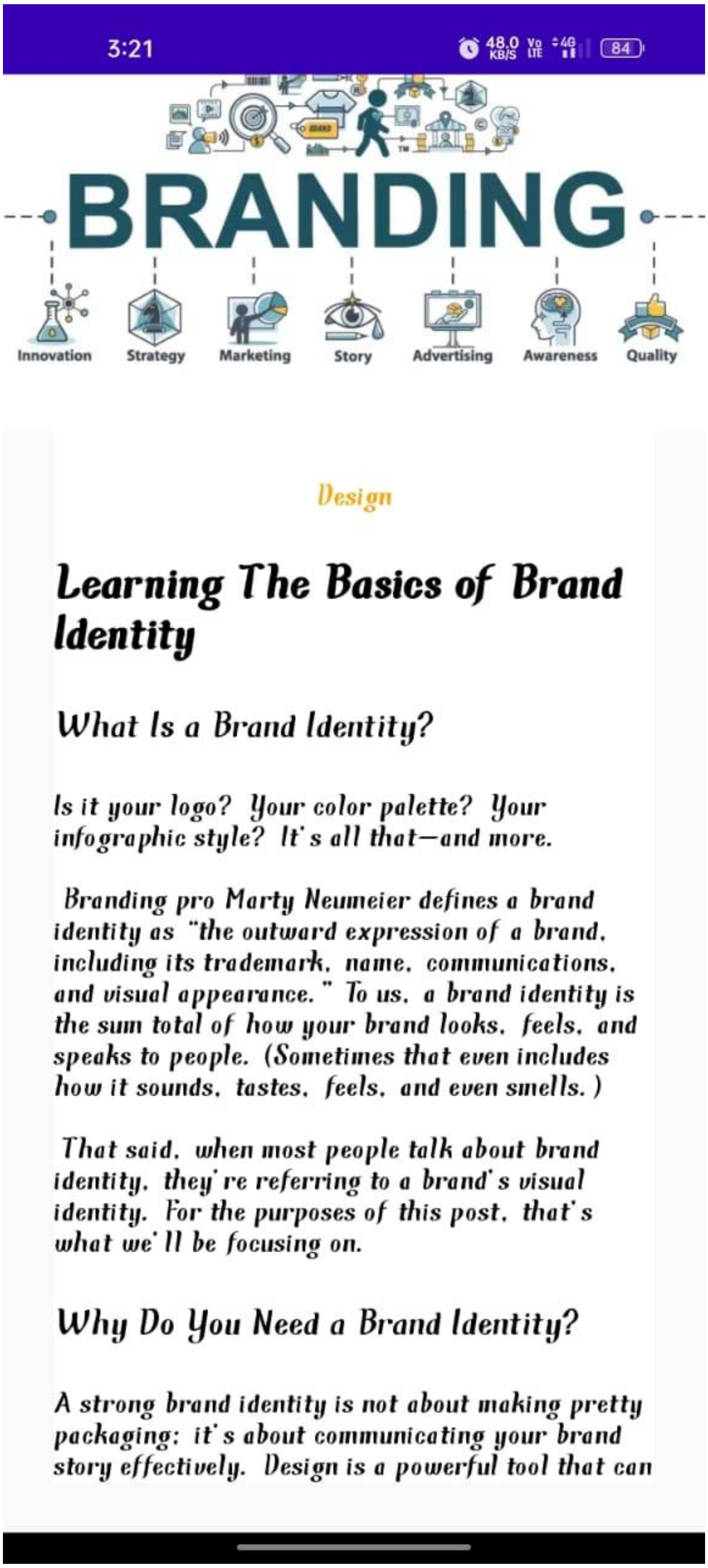


Architecture

City Phenomenon between Urban Structure and Composition

Abstract

Cities are not just a sum of buildings, but especially a set of social relations that their inhabitants develop. Cities are characterized by a wide variety of social groups and lifestyles. An urban composition represents a form of the city in which it gets a formal order, so that the shape of any urban ensemble is not linked to a random phenomenon, but to an intervention mastered and understood as such. For the city, the urban composition represents what the architectural composition represents for a building. This concept regarding the composition is common both to the architecture and to the city. The main property of the composition is that it transforms a possibly dispersed ensemble into a whole, resolving the contradictions that arise when the requirements and conditions of the project are numerous. Spatial forms and urban compositions are built over time, longer than that of architectural composition. On the other hand, "design of the urban environment" is understood by us as a complex formation of public spaces of the city, located on the ground floor level of the city building and ensuring the vital activity of the urban community. This chapter will study the city phenomenon on a large scale.



Design

Learning The Basics of Brand Identity

What Is a Brand Identity?

Is it your logo? Your color palette? Your infographic style? It's all that—and more.

Branding pro Marty Neumeier defines a brand identity as "the outward expression of a brand, including its trademark, name, communications, and visual appearance." To us, a brand identity is the sum total of how your brand looks, feels, and speaks to people. (Sometimes that even includes how it sounds, tastes, feels, and even smells.)

That said, when most people talk about brand identity, they're referring to a brand's visual identity. For the purposes of this post, that's what we'll be focusing on.

Why Do You Need a Brand Identity?

A strong brand identity is not about making pretty packaging; it's about communicating your brand story effectively. Design is a powerful tool that can

4. ADVANTAGES & DISADVANTAGES

Advantages

- Modern and user-friendly design: By incorporating Material Design and OWL application framework, the study app can have a modern and user-friendly design, which can enhance the user experience and encourage users to engage with the app.

- Personalization and adaptability: The OWL application framework provides personalization and adaptability features, allowing users to tailor the app to their individual learning preferences and needs.

- Interactive and engaging learning experience: The OWL application framework also supports interactive and engaging learning experiences, such as gamification, collaboration, and multimedia content, which can increase user engagement and motivation.

- Open source and community support: The OWL application framework is open source and has a supportive community of developers and educators, which can provide valuable resources and support for app development.

- Scalability and flexibility: The OWL application framework is scalable and flexible, allowing for the addition of new features and functionality as the study app grows and evolves.

Disadvantages

- Complexity: Using both Material Design and the OWL application framework can add complexity to the development process, which may require more time and resources to develop and maintain the app.

- Learning curve: Developers and educators may need to invest time and effort to learn how to use the OWL application framework and properly incorporate it into the study app.

- Limited user base: While the OWL application framework has a supportive community of developers and educators, it may have a limited user base compared to more widely used educational technology systems.

- Compatibility issues: Integrating the study app with existing educational technology systems, such as learning management systems, may require compatibility testing and additional development work.

- Security concerns: As with any app, security concerns must be addressed, such as protecting user data and preventing unauthorized access.

5. APPLICATIONS

- eLearning: The study app can be used as an eLearning tool, providing interactive and engaging educational content to learners of all ages and backgrounds.

- Corporate training: The app can be used as a corporate training tool, providing personalized and adaptable learning experiences to employees in a variety of industries.

- Language learning: The study app can be used as a language learning tool, incorporating gamification, multimedia content, and social learning features to make language learning more engaging and effective.

- Test preparation: The app can be used as a test preparation tool, providing learners with practice questions, quizzes, and interactive learning resources to help them prepare for standardized tests and exams.

- Professional development: The study app can be used as a professional development tool, providing learners with access to courses, workshops, and other learning opportunities to help them advance their careers and gain new skills.

- Home schooling: The app can be used as a home schooling tool, providing parents and children with a comprehensive and interactive educational resource that can be adapted to meet individual learning needs.

6. CONCLUSION

In conclusion, developing a study app using both the OWL application framework and Material Design in Android Studio can offer various advantages, including a modern and user-friendly design, personalization and adaptability, interactive and engaging learning experiences, open source and community support, scalability and flexibility, and integration with learning management systems.

However, it is also important to consider potential disadvantages, such as complexity, learning curve, limited user base, compatibility issues, security concerns, and maintenance requirements.

Overall, a study app developed using both the OWL application framework and Material Design in Android Studio has diverse applications in eLearning, corporate training, language learning, test preparation, professional development, home schooling, and more. It can provide learners with access to interactive and engaging educational resources that can be tailored to meet their individual learning needs, making it a versatile tool for learning and professional development.

7. FUTURE SCOPE

- Social learning: The study app can leverage social learning features, such as peer-to-peer collaboration, discussion forums, and social media integration, to enable learners to connect and learn from each other.

- Mobile learning: Mobile learning is becoming increasingly popular, and the study app can be optimized for mobile devices, providing learners with anytime, anywhere access to educational content and resources.

- Adaptive learning: The study app can use data analytics and learner performance data to personalize learning experiences and adapt to individual learner needs and preferences.

Overall, the future scope of a study app developed using both the OWL application framework and Material Design in Android Studio is vast and varied, with numerous opportunities to enhance learning experiences and improve educational outcomes. By incorporating emerging technologies and trends, the study app can continue to evolve and adapt to the changing needs and preferences of learners in the future.

8. APPENDIX

A. Source code

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/andr
oid"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"

android:dataExtractionRules="@xml/data_extraction_rul
es"
        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>
```

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.PasswordVisualTransfor
mation
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.OwlApplicationT
heme

class LoginActivity : ComponentActivity() {
    private lateinit var databaseHelper:
UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(this)
        setContent {
            LoginScreen(this, databaseHelper)
        }
    }
}
@Composable
fun LoginScreen(context: Context, databaseHelper:
UserDatabaseHelper) {
    var username by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var error by remember { mutableStateOf("") }

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

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

        Text(
            fontSize = 36.sp,
            fontWeight = FontWeight.ExtraBold,
            fontFamily = FontFamily.Cursive,
            text = "Login"
        )
        Spacer(modifier = Modifier.height(10.dp))

        TextField(
            value = username,
            onValueChange = { username = it },
            label = { Text("Username") },
            modifier = Modifier.padding(10.dp)
                .width(280.dp)
        )
        TextField(
            value = password,
            onValueChange = { password = it },
            label = { Text("Password") },
            visualTransformation =
PasswordVisualTransformation(),
            modifier = Modifier.padding(10.dp)
                .width(280.dp)
        )

        if (error.isNotEmpty()) {
            Text(
                text = error,
                color = MaterialTheme.colors.error,
                modifier = Modifier.padding(vertical = 16.dp)
            )
        }

        Button(
            onClick = {
                if (username.isNotEmpty() &&
password.isNotEmpty()) {
                    val user =
databaseHelper.getUserByUsername(username)
                    if (user != null && user.password ==
password) {
                        error = "Successfully log in"
                        context.startActivity(
                            Intent(
                                context,
                                MainActivity::class.java
                            )
                        )
                        //onLoginSuccess()
                    }
                    else {
                        error =  "Invalid username or password"
                    }

                } else {
                    error = "Please fill all fields"
                }
            },
            modifier = Modifier.padding(top = 16.dp)
        ) {
            Text(text = "Login")
        }
        Row {
            TextButton(onClick = {context.startActivity(
                Intent(
                    context,
                    RegisterActivity::class.java
                )
            )}
            )
            { Text(text = "Register") }
            TextButton(onClick = {
            })

            {
                Spacer(modifier = Modifier.width(60.dp))
                Text(text = "Forget password?")
            }
        }
    }
}
private fun startMainPage(context: Context) {
    val intent = Intent(context, MainActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
```

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.OwlApplicationT
heme

class MainActivity2 : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Greeting()
        }
    }
}
@Composable
fun Greeting() {
    Column(
        modifier = Modifier.padding(start = 26.dp, end =
26.dp, bottom = 26.dp)
            .verticalScroll(rememberScrollState())
            .background(Color.White),
        verticalArrangement = Arrangement.Top
    ) {

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

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

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

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

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

        )
        Spacer(modifier = Modifier.height(20.dp))
        Text(
            text = stringResource(id =
R.string.subheading1_1),
            modifier = Modifier.align(Alignment.Start),
            fontSize = 20.sp
        )

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

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

        Spacer(modifier = Modifier.height(20.dp))
        Text(
            text = stringResource(id =
R.string.subheading1_2),
            modifier = Modifier.align(Alignment.Start),
            fontSize = 20.sp
        )

        Spacer(modifier = Modifier.height(20.dp))
        Text(
            text = stringResource(id = R.string.text1_2),
            modifier = Modifier.align(Alignment.Start),
            textAlign = TextAlign.Justify,
            fontSize = 16.sp
        )




    }
}
```

MainActivity3.kt

```
package com.example.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.OwlApplicationT
heme

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
        )




    }
}
```

MainActivity.5.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.OwlApplicationT
heme

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.PasswordVisualTransfor
mation
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.OwlApplicationT
heme

class RegisterActivity : ComponentActivity() {
    private lateinit var databaseHelper:
UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(this)
        setContent {
            RegistrationScreen(this, databaseHelper)
        }
    }
}

@Composable
fun RegistrationScreen(context: Context,
databaseHelper: UserDatabaseHelper) {

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

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

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

        Text(
            fontSize = 36.sp,
            fontWeight = FontWeight.ExtraBold,
            fontFamily = FontFamily.Cursive,
            text = "Register"
        )
        Spacer(modifier = Modifier.height(10.dp))
        TextField(
            value = username,
            onValueChange = { username = it },
            label = { Text("Username") },
            modifier = Modifier
                .padding(10.dp)
                .width(280.dp)

        )

        TextField(
            value = email,
            onValueChange = { email = it },
            label = { Text("Email") },
            modifier = Modifier
                .padding(10.dp)
                .width(280.dp)
        )

        TextField(
            value = password,
            onValueChange = { password = it },
            label = { Text("Password") },
            visualTransformation =
PasswordVisualTransformation(),
            modifier = Modifier
                .padding(10.dp)
                .width(280.dp)
        )
        if (error.isNotEmpty()) {
            Text(
                text = error,
                color = MaterialTheme.colors.error,
                modifier = Modifier.padding(vertical = 16.dp)
            )
        }

        Button(
            onClick = {
                if (username.isNotEmpty() &&
password.isNotEmpty() && email.isNotEmpty()) {
                    val user = User(
                        id = null,
                        firstName = username,
                        lastName = null,
                        email = email,
                        password = password
                    )
                    databaseHelper.insertUser(user)
                    error = "User registered successfully"
                    // Start LoginActivity using the current
context
                    context.startActivity(
                        Intent(
                            context,
                            LoginActivity::class.java
                        )
                    )
                } else {
                    error = "Please fill all fields"
                }
            },
            modifier = Modifier.padding(top = 16.dp)
        ) {
            Text(text = "Register")
        }
        Spacer(modifier = Modifier.width(10.dp))
        Spacer(modifier = Modifier.height(10.dp))

        Row() {
            Text(
                modifier = Modifier.padding(top = 14.dp),
text = "Have an account?"
            )
            TextButton(onClick = {
                context.startActivity(
                    Intent(
                        context,
                        LoginActivity::class.java
                    )
                )
            })

            {
                Spacer(modifier = Modifier.width(10.dp))
                Text(text = "Log in")
            }
        }
```

```
    }
}
private fun startLoginActivity(context: Context) {
    val intent = Intent(context, LoginActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
```

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?,

    )
```

app/src/main/java/com/example/owlapplication/UserDa
o.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)
}
```

app/src/main/java/com/example/owlapplication/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()
   }
```

```
    @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_FI
RST_NAME)),
                lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_L
AST_NAME)),
                email =
cursor.getString(cursor.getColumnIndex(COLUMN_E
MAIL)),
                password =
cursor.getString(cursor.getColumnIndex(COLUMN_P
ASSWORD)),
            )
        }
        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_FI
RST_NAME)),
                lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_L
AST_NAME)),
                email =
cursor.getString(cursor.getColumnIndex(COLUMN_E
MAIL)),
                password =
cursor.getString(cursor.getColumnIndex(COLUMN_P
ASSWORD)),
            )
        }
        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_FI
RST_NAME)),
                    lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_L
AST_NAME)),
                    email =
cursor.getString(cursor.getColumnIndex(COLUMN_E
MAIL)),
                    password =
cursor.getString(cursor.getColumnIndex(COLUMN_P
ASSWORD)),
                )
                users.add(user)
            } while (cursor.moveToNext())
        }
        cursor.close()
        db.close()
        return users
    }

}
```

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
    )
    */
)
```