CONTENTS

INTRODUCTION

OWL is a material design study app designed to help students organize, manage, and optimize their study routines. The app comes with a user-friendly interface and a range of features that make it easy for students to stay on top of their studies, set goals, and track progress.

1.1 Overview:

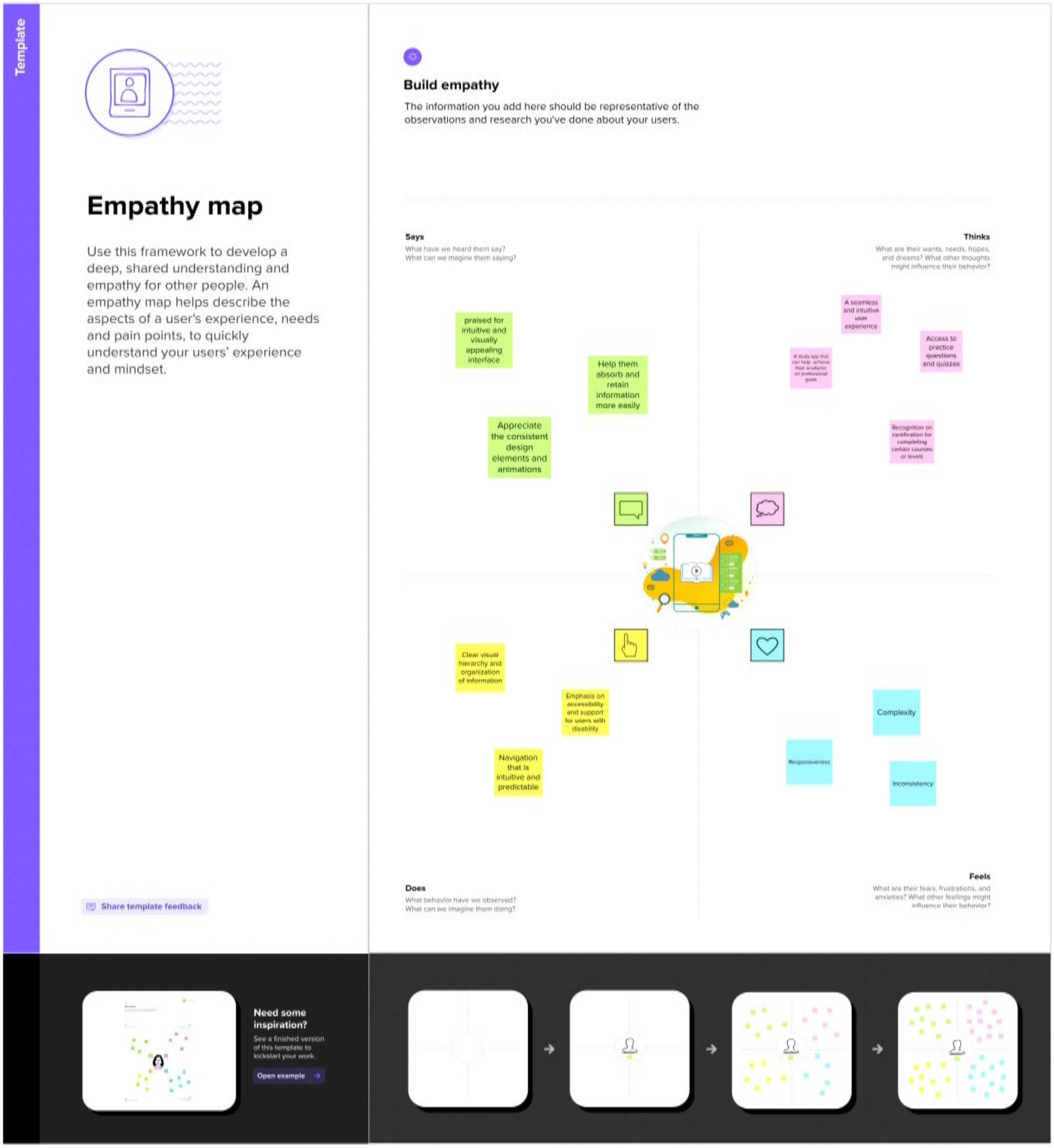
- OWL is a study app that is designed to help students organize their study routines effectively. It comes with a user-friendly material design interface and a range of features that make it easy to plan, manage, and optimize studying.
- The app allows users to customize their study plans based on their personal preferences and schedules. It also enables students to set reminders and notifications for studying, helping them stay on track and meet deadlines easily.
- In addition, OWL comes with built-in analytics tools that allow students to track their progress and evaluate their learning, making it easier for them to make necessary adjustments to their study routine.

1.2 Purpose:

- Furthermore, the app provides a range of tools and resources to help students better manage their time, stay motivated, and improve their academic performance.
- Overall, the main purpose of OWL is to support students in achieving their academic goals and enhancing their learning experience through effective study planning and implementation.

PROBLEM DEFINITION & DESIGN THINKING

2.1 Empathy Map:



Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

Build empathy

The information you add here should be representative of the observations and research you've done about your users.

Says

What have we heard them say? What can we magine them saying?

praised for intuitive and visually appealing interface

Help them absorb and retain information more easily

Appreciate the consistent design elements and animations

Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

A seamless and intuitive user experience

A study app that can help achieve their academic on professional goals

Access to practice questions and quizzes

Recognition on certification for completing certain courses or levels

Clear visual hierarchy and organization of information

Emphasis on accessibility and support for users with disability

Navigation that is intuitive and predictable

Complexity

Responsiveness

Inconsistency

Does

What behavior have we observed? What can we imagine them doing?

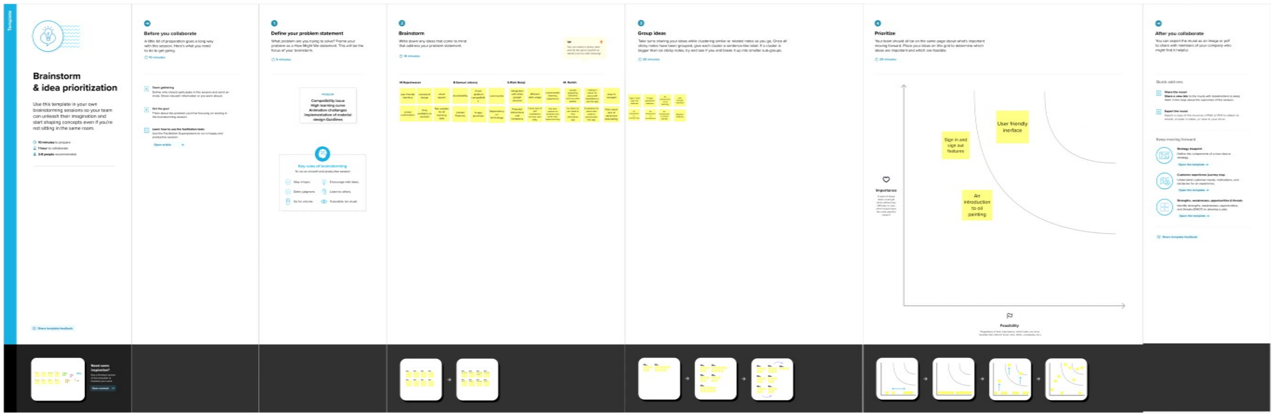
Feels

What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

Need some inspiration?

See a finished version of this template to kickstart your work.

2.2 Ideation & 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

Need some inspiration?
See a finished version of this template to kickstart your work.
Open example

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

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

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

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
Compatibility issue
High learning curve
Animation challenges
Implementation of material design Guidlines

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

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

M.Rajeshwaran
user friendly interface
consistent design
visual appeal
limited customization
Only available on android
Not suitable for all learning style

B.Samuel Jebaraj
Accessibility
Cross platform compatibility
Community
Limited features
In-app purchase
Dependency on technology

S.Rishi Balaji
Integration with other google services
Efficient data usage
customizable learning experience
Potential distractions and Careplans

M. Rohith
easy to navigate
They cause lack of advanced note-taking

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

Sign in and sign out features
An introduction to oil painting
Regular features

4 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

Sign in and sign out features
User friendly inerface
An introduction to oil painting
Importance
Feasibility

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 →

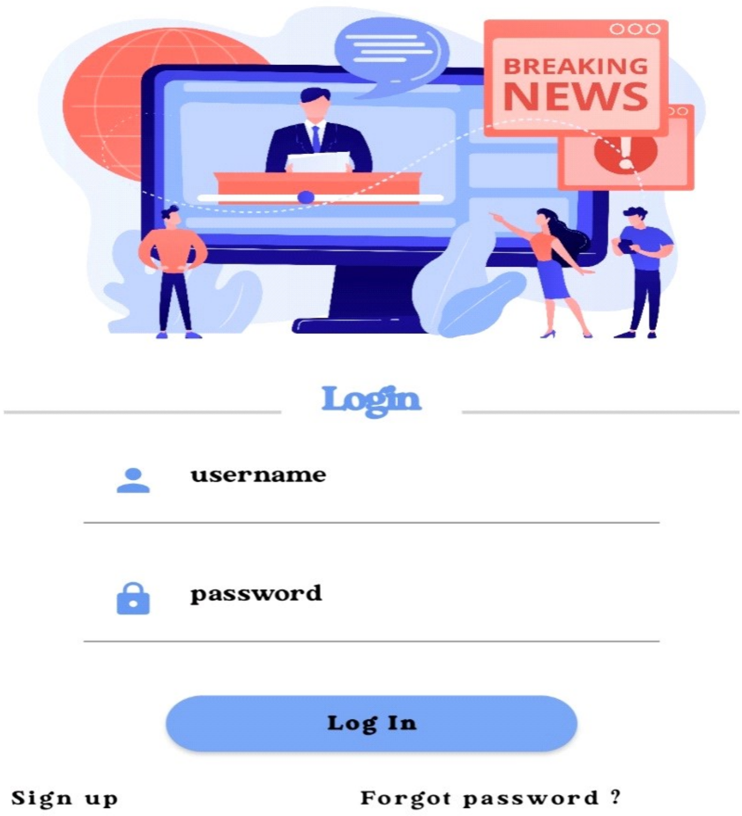
Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

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

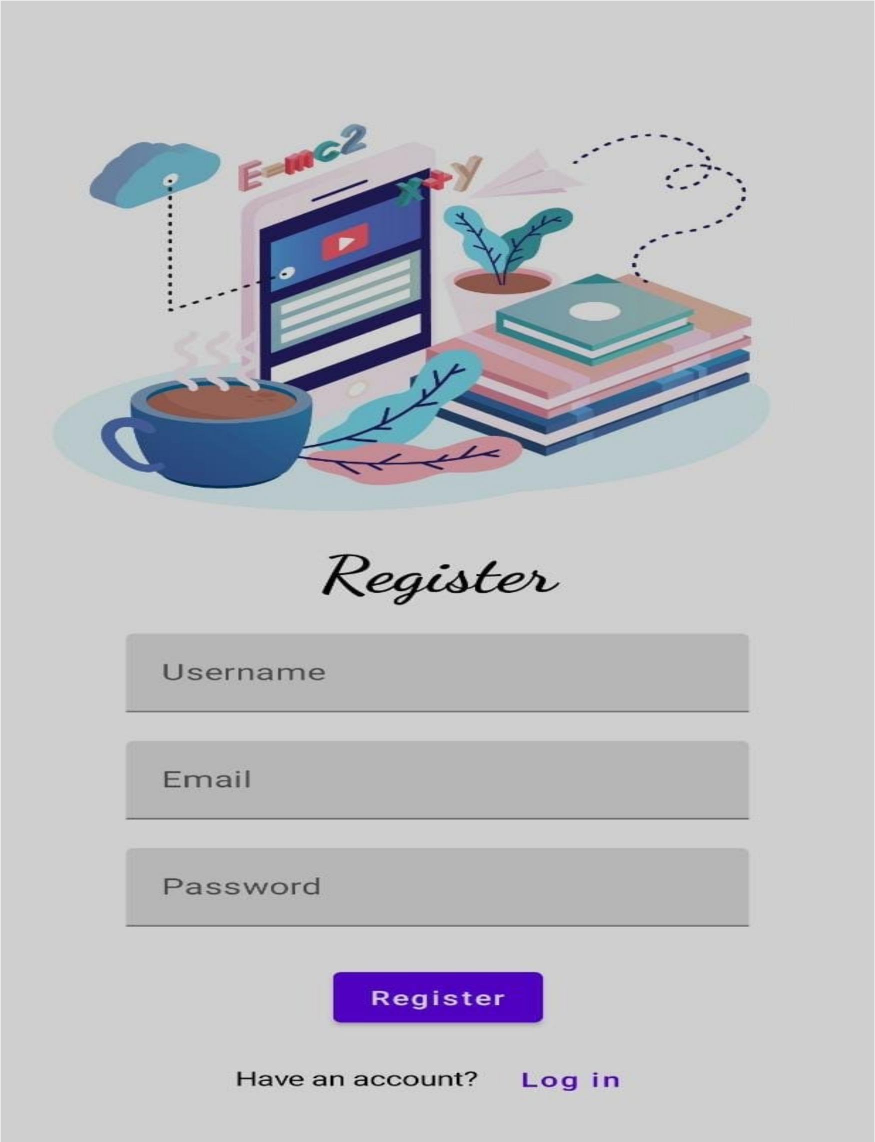
Share template feedback

RESULT

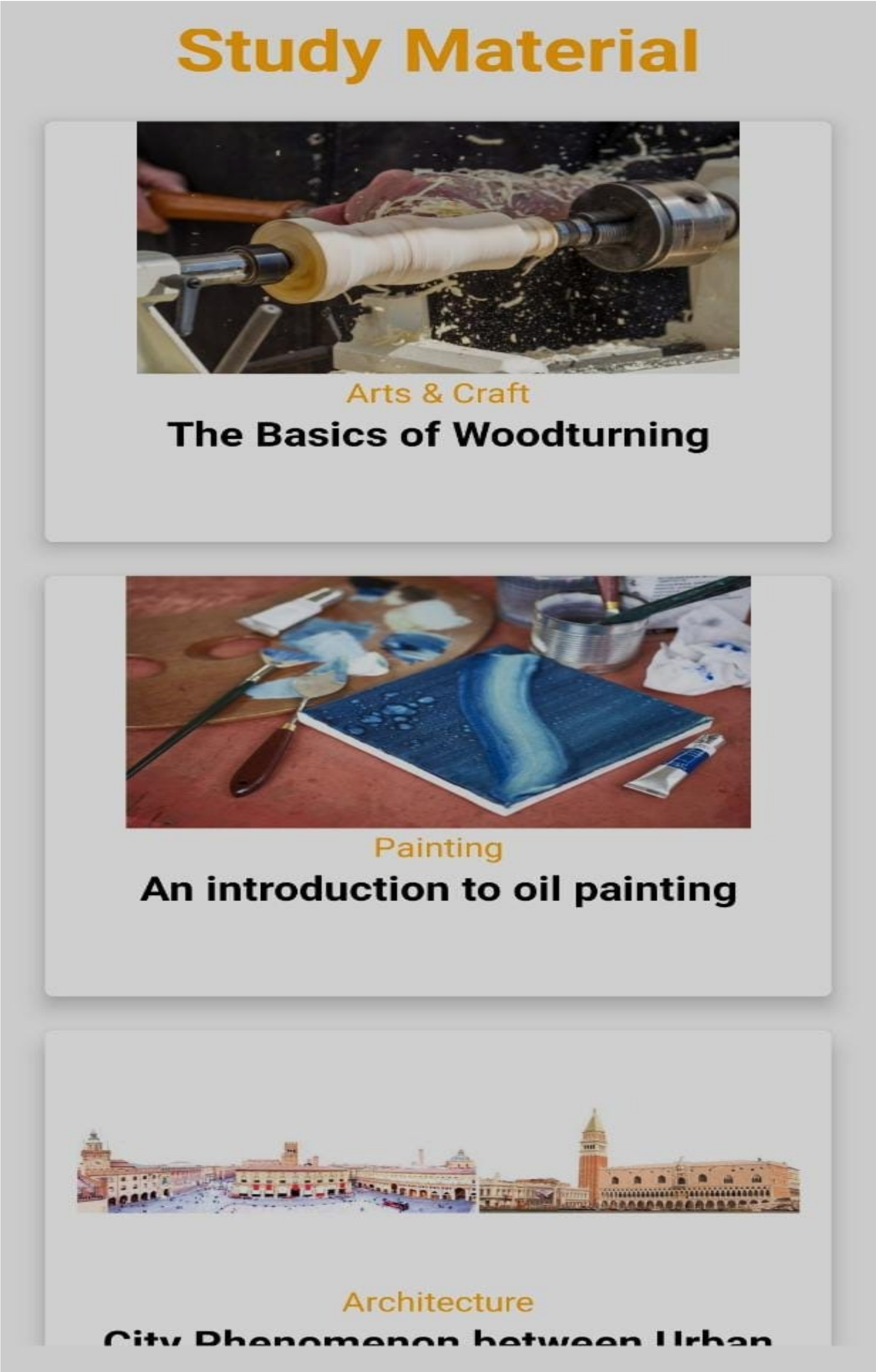
Sign in Page:



Sign up page:


E=mc2
x+y
Register
Username
Email
Password
Register
Have an account?
Log in

Home Page:



Study Material

Arts & Craft

The Basics of Woodturning

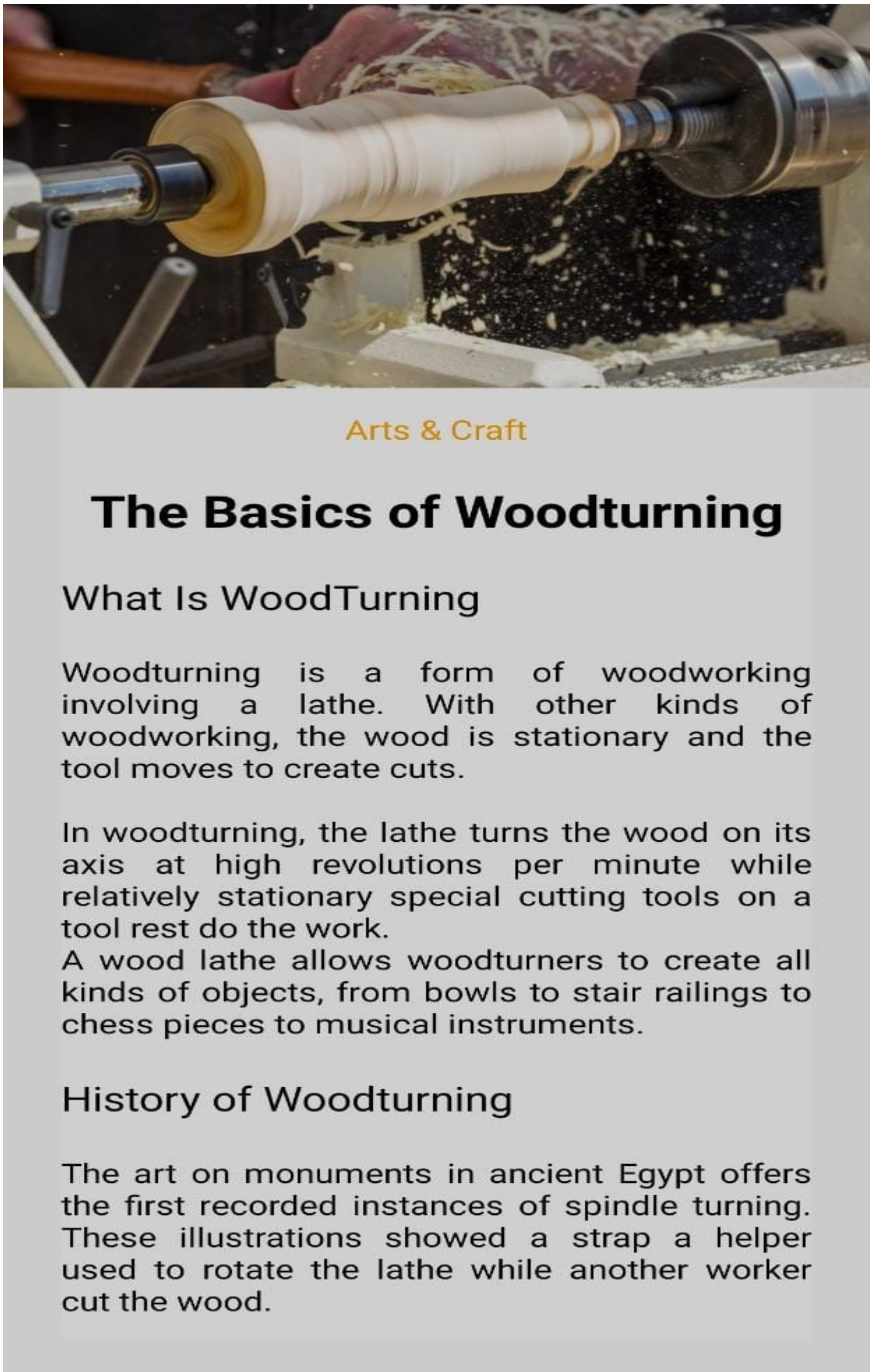
Painting

An introduction to oil painting

Architecture

City Phenomenon between Urban

Arts and craft:



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.

ADVANTAGES & DISADVANTAGES

- **Advantages:**

  - Comprehensive study planning: OWL enables users to organize study routines, set goals, track progress, and optimize their learning experience with ease. The app provides a range of customizable features and tools to help students manage their time efficiently and stay on track.
  - Intuitive and user-friendly interface: OWL is designed with a modern and intuitive interface that makes it easy for students to navigate and access all the necessary features and functionality.
  - Material design: The app's material design principles ensure that users have a visually pleasing and cohesive experience while using the app.
  - Motivational tools: OWL offers a range of motivational tools such as progress tracking,...

- **Disadvantages:**

  - One potential disadvantage of OWL as a material design study app is that it may not be suitable for all learning styles.
  - While the app is designed to accommodate different learning preferences, such as visual or auditory, some students may find that the features and tools provided by OWL do not align with their individual learning needs.
  - Additionally, the app may not be effective in providing personalized feedback or addressing specific learning challenges that students may face.

APPLICATIONS

- Personalized learning: OWL allows students to personalize their study sessions by choosing the topics they want to focus on, and the pace at which they want to learn.
- Visual and auditory learning: The app provides visual aids and audio instructions to cater to different learning styles.
- Interactive learning: OWL offers interactive quizzes and assessments that allow students to practice and test their knowledge.
- Time management: The app provides a scheduling feature that helps students organize their study sessions efficiently.
- Teacher support: OWL can also be used by educators to create and share course materials, assessments, and other resources with their students, as well as track and monitor student progress.
- Accessibility: The app is designed to be accessible to students with disabilities, offering features such as screen reading software compatibility, adjustable font sizes, and closed captions.
- Distance learning: With the increasing popularity of online education and remote learning, OWL can be an excellent tool for students who need to study from home or in different locations.

Conclusion

Overall, OWL is an excellent material design study app that offers a range of features and benefits for students of all ages and abilities. From its intuitive and user-friendly interface to its engaging and interactive study tools, OWL is designed to help students learn more effectively and efficiently. Whether you are studying for exams, working on projects, or simply trying to improve your knowledge and skills, OWL can help you achieve your goals and succeed in your studies. So if you are looking for a powerful and versatile study app that can help you on your learning journey, OWL is definitely worth checking out.

Future Scopes

Introducing new study features: OWL can continue to add new features and study tools to enhance the learning experience of its users. For example, it could include interactive quizzes, flashcards, and virtual study sessions.Expanding the subject areas: At present, OWL primarily focuses on science and maths subjects, but it can expand into other subject areas like literature, geography, history, and language learning to make it more comprehensive.Integration of AI: Incorporating AI technologies can help OWL personalize study recommendations based on the user's past performance and learning patterns. This can...

APPENDIX

Source Code:

https://github.com/rajeshwaran10/Study_Material_App

Code:-

1. LoginActivity.kt

```
package com.example.owlapplication


import android.content.Context

import android.content.Intent

import android.os.Bundle

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.Image

import androidx.compose.foundation.background

import androidx.compose.foundation.layout.*

import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat
import com.example.owlapplication.ui.theme.OwlApplicationTheme

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

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

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

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

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

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

TextField(
  value = username,
  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)
}
```

2. 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,
            )
        }
    }


}
```

```
}
```

3. 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
        )



    }
}
```

4. 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
        )
    }
}
```

5. MainActivity4.kt

```
package com.example.owlapplication


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

class MainActivity4 : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Greeting2()
        }
    }
}
@Composable
fun Greeting2() {
    Column(
        modifier = Modifier.padding(start = 26.dp, end = 26.dp, bottom = 26.dp)
            .verticalScroll(rememberScrollState())
            .background(Color.White),
        verticalArrangement = Arrangement.Top
    ) {
```

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

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

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

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

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

```
)
Spacer(modifier = Modifier.height(20.dp))
Text(
    text = stringResource(id = R.string.subheading3_1),
    modifier = Modifier.align(Alignment.Start),
    fontSize = 20.sp
)

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

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

Spacer(modifier = Modifier.height(20.dp))
Text(
    text = stringResource(id = R.string.subheading3_2),
    modifier = Modifier.align(Alignment.Start),
    fontSize = 20.sp
```

```
        )

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

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

    }
}
```

6. MainActivity5.kt

```
package com.example.owlapplication

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

class MainActivity5 : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Greeting3()
        }
    }
}
@Composable
fun Greeting3() {
    Column(
        modifier = Modifier.padding(start = 26.dp, end = 26.dp, bottom = 26.dp)
            .verticalScroll(rememberScrollState())
            .background(Color.White),
        verticalArrangement = Arrangement.Top
    ) {
```

```
Image(
    painterResource(id = R.drawable.img_4),
    contentDescription = "",
    modifier = Modifier.align(Alignment.CenterHorizontally)
        .scale(scaleX = 1.5F, scaleY = 1.5F)
)

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

Text(
    text = stringResource(id = R.string.course4),
    color = Color(0xFFFFA500),
    fontSize = 16.sp,
    modifier = Modifier.align(Alignment.CenterHorizontally)
)

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

Text(
    text = stringResource(id = R.string.topic4),
    fontWeight = FontWeight.Bold,
    fontSize = 26.sp,
    modifier = Modifier.align(Alignment.CenterHorizontally)
```

```
)
Spacer(modifier = Modifier.height(20.dp))
Text(
    text = stringResource(id = R.string.subheading4_1),
    modifier = Modifier.align(Alignment.Start),
    fontSize = 20.sp
)

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

Text(
    text = stringResource(id = R.string.text4_1),
    modifier = Modifier.align(Alignment.Start),
    textAlign = TextAlign.Justify,
    fontSize = 16.sp
)

Spacer(modifier = Modifier.height(20.dp))
Text(
    text = stringResource(id = R.string.subheading4_2),
    modifier = Modifier.align(Alignment.Start),
    fontSize = 20.sp
```

```
            )

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

            Text(
                text = stringResource(id = R.string.text4_2),
                modifier = Modifier.align(Alignment.Start),
                textAlign = TextAlign.Justify,
                fontSize = 16.sp
            )



        }
    }
```

7. RegisterActivity.kt

```
package com.example.owlapplication


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

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

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

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

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

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

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

  Spacer(modifier = Modifier.height(10.dp))
  TextField(
    value = username,
    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)
}
```