PROJECT TITLE

OWL – M: A MATERIAL DESIGN STUDY APP

DESIGNED BY:

SRI PARAMAKALYANI COLLEGE (CODE-123),

ALWARKURICHI-627416

DEPARTMENT OF COMPUTER APPLICATION

MENTOR

E.JACQULINE, M.C.A., M.Phil.,

TEAM

TEAM LEADER

S.VANAJA (20201231401247)

TEAM MEMBERS

M.BHARATHI (20201231401207)

U.ESAKKIAMMAL (20201231401211)

L.MUTHUMEENA (20201231401224)

PROJECT INDEX

CHAPTER-I INTRODUCTION

1.1 Overview

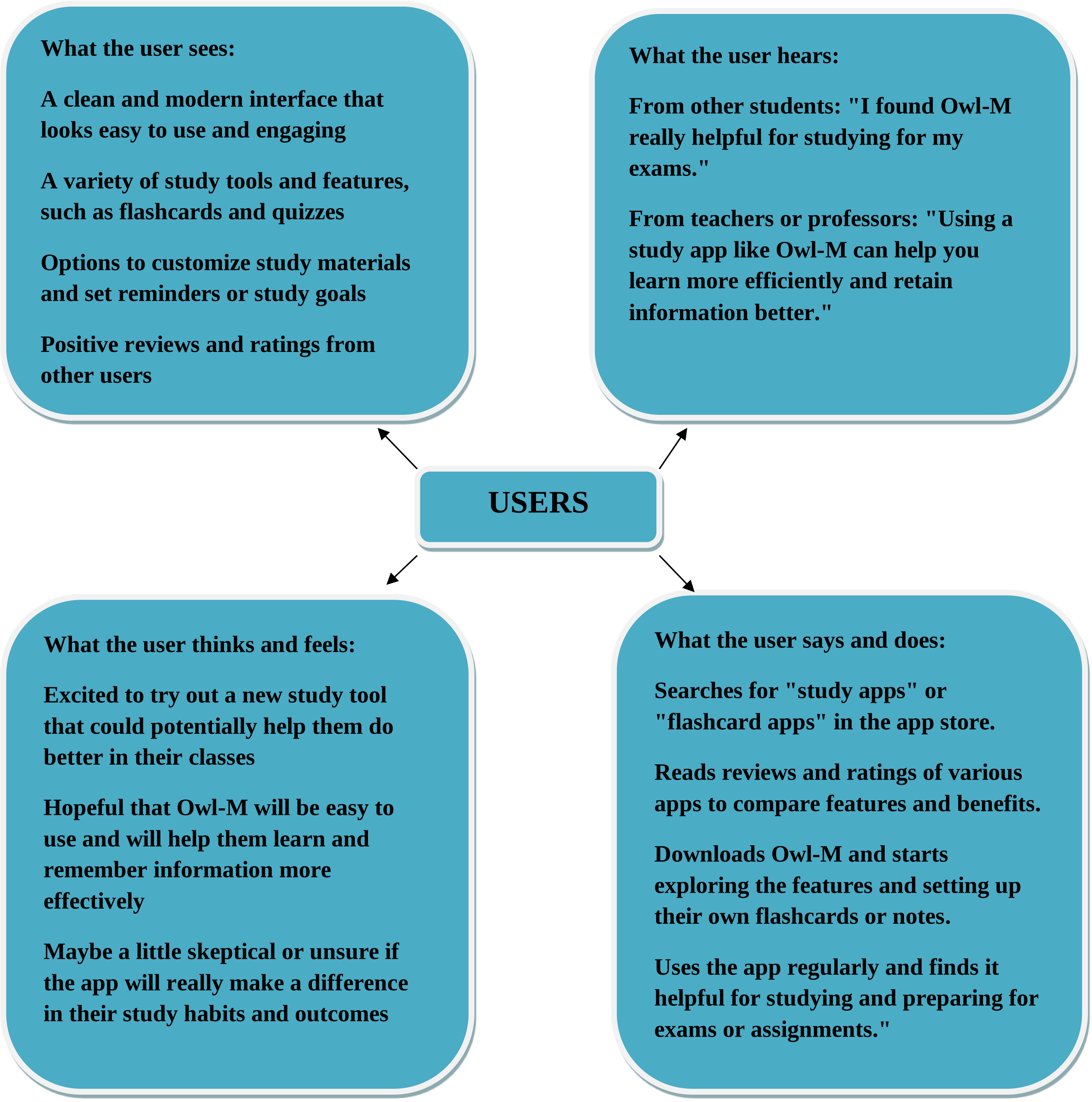
- Owl-M is a material design study app designed to help users learn and retain information efficiently. The app has a sleek and modern design, with a focus on usability and simplicity.
- The main feature of Owl-M is its flashcard system, which allows users to create and study custom decks of flashcards. Users can add images and audio to their flashcards, as well as organize them into folders for easy access.
- Another feature of Owl-M is its quizzes, which can be used to test knowledge and track progress. Users can choose from a variety of quiz types, including multiple choice, matching, and fill-in-the-blank.
- Owl-M also includes a built-in note-taking system, allowing users to take and organize notes on any topic. The app also offers the ability to set reminders and study goals, making it easy to stay on track and motivated.
- Overall, Owl-M is a powerful study tool that combines intuitive design with useful features, making it an excellent choice for students of all ages and levels."

1.2 Purpose

- The purpose of Owl-M is to provide users with a powerful and efficient tool for studying and retaining information. The app is designed to help users learn and memorize new concepts, vocabulary, and other types of information by using a variety of study techniques, such as flashcards and quizzes.
- Owl-M's material design and user-friendly interface aim to make studying more enjoyable and accessible, allowing users to focus on their learning goals without feeling overwhelmed or distracted by complicated features or interfaces.
- The app's note-taking system, study goal tracking, and reminder features also help users stay organized and motivated, making it easier to stay on track with their learning objectives.
- Overall, the purpose of Owl-M is to help users achieve their academic and personal learning goals by providing them with a reliable and effective study app that simplifies the learning process and promotes better retention and understanding of information.
- The app is designed to be user-friendly and intuitive, with a focus on simplicity and usability. Its main goal is to help users study smarter, not harder, by providing a range of features and tools that make studying more effective and enjoyable. These features include customizable flashcards, quizzes, note-taking, reminders, and study goals. By using Owl-M, users can improve their learning outcomes and achieve their academic goals.

CHAPTER-II PROBLEM DEFINITION&DESIGN THINKING

2.1 Empathy Map



USERS

What the user sees:

A clean and modern interface that looks easy to use and engaging

A variety of study tools and features, such as flashcards and quizzes

Options to customize study materials and set reminders or study goals

Positive reviews and ratings from other users

What the user hears:

From other students: "I found Owl-M really helpful for studying for my exams."

From teachers or professors: "Using a study app like Owl-M can help you learn more efficiently and retain information better."

What the user thinks and feels:

Excited to try out a new study tool that could potentially help them do better in their classes

Hopeful that Owl-M will be easy to use and will help them learn and remember information more effectively

Maybe a little skeptical or unsure if the app will really make a difference in their study habits and outcomes

What the user says and does:

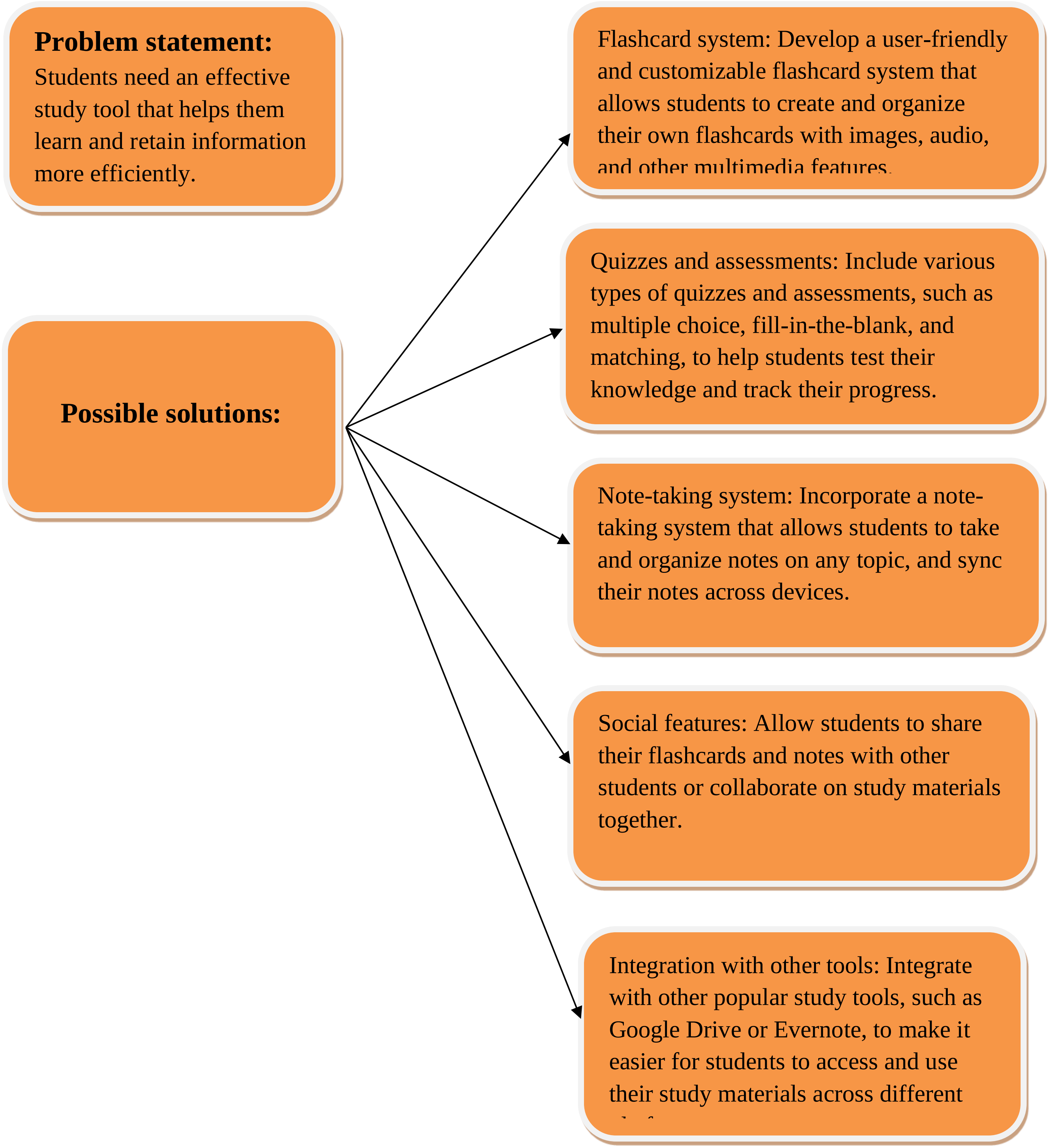
Searches for "study apps" or "flashcard apps" in the app store.

Reads reviews and ratings of various apps to compare features and benefits.

Downloads Owl-M and starts exploring the features and setting up their own flashcards or notes.

Uses the app regularly and finds it helpful for studying and preparing for exams or assignments."

2.2 Ideation & Brainstorming Map



Problem statement: Students need an effective study tool that helps them learn and retain information more efficiently.

Possible solutions:

Flashcard system: Develop a user-friendly and customizable flashcard system that allows students to create and organize their own flashcards with images, audio, and other multimedia features.

Quizzes and assessments: Include various types of quizzes and assessments, such as multiple choice, fill-in-the-blank, and matching, to help students test their knowledge and track their progress.

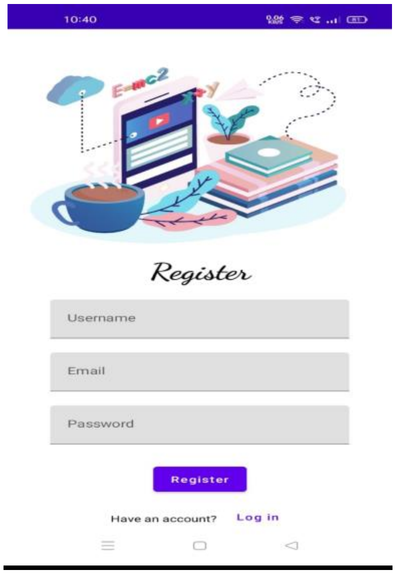
Note-taking system: Incorporate a note-taking system that allows students to take and organize notes on any topic, and sync their notes across devices.

Social features: Allow students to share their flashcards and notes with other students or collaborate on study materials together.

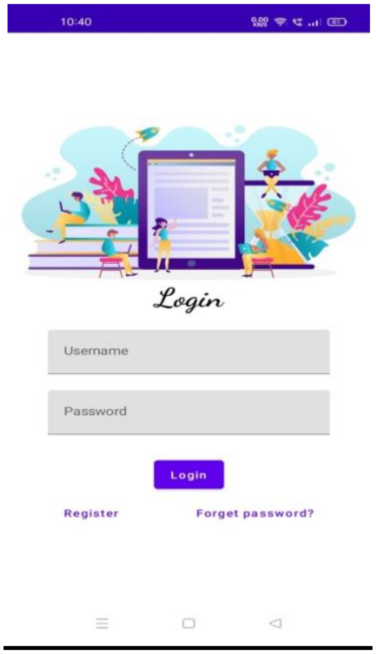
Integration with other tools: Integrate with other popular study tools, such as Google Drive or Evernote, to make it easier for students to access and use their study materials across different

CHAPTER-III RESULT

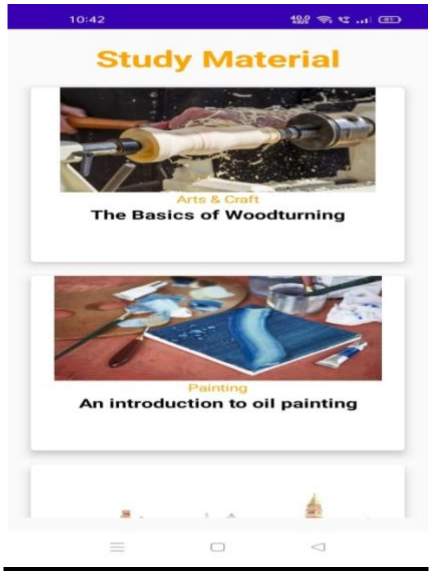
REGISTER PAGE



LOGIN



MAINPAGE



10:42

Study Material

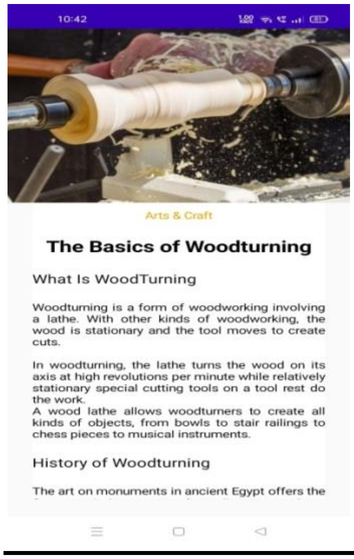
Arts & Craft

The Basics of Woodturning

Painting

An introduction to oil painting

BOOKPAGE



10:42

Arts & Craft

The Basics of Woodturning

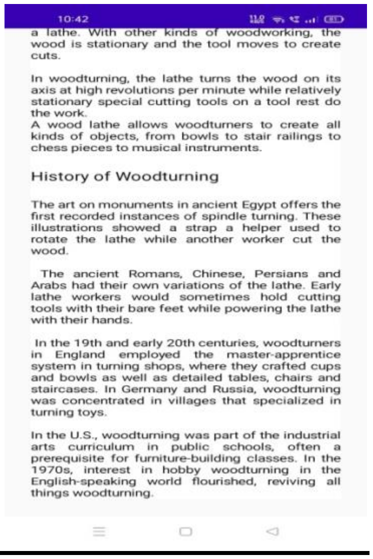
What Is WoodTurning

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



10:42

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.

The ancient Romans, Chinese, Persians and Arabs had their own variations of the lathe. Early lathe workers would sometimes hold cutting tools with their bare feet while powering the lathe with their hands.

In the 19th and early 20th centuries, woodturners in England employed the master-apprentice system in turning shops, where they crafted cups and bowls as well as detailed tables, chairs and staircases. In Germany and Russia, woodturning was concentrated in villages that specialized in turning toys.

In the U.S., woodturning was part of the industrial arts curriculum in public schools, often a prerequisite for furniture-building classes. In the 1970s, interest in hobby woodturning in the English-speaking world flourished, reviving all things woodturning.

CHAPTER-IV ADVANTAGES &DISADVANTAGES

4.1 ADVANTAGES

- **Customizable flashcards:** Owl-M's flashcard system allows users to create and customize their own flashcards with images and audio, making it easier to remember and recall information.
- **Quizzes and assessments:** The app offers various types of quizzes and assessments to help users test their knowledge and track their progress.
- **User-friendly interface:** Owl-M's material design and user-friendly interface make it easy and enjoyable to use, even for students who may not be tech-savvy.
- **Note-taking system:** The app's note-taking system allows users to take and organize notes on any topic, and sync their notes across devices.
- **Goal setting and reminders:** The app offers features that allow users to set study goals and reminders, helping them stay on track and motivated to achieve their learning objectives.

4.2 DISADVANTAGES

- **Limited free features:** While the app is free to download, some features, such as the ability to create unlimited flashcards or access advanced analytics, require a paid subscription.
- **Limited social features:** The app currently does not offer much in the way of social features, such as the ability to collaborate with other students or share study materials.
- **Requires self-discipline:** Like any study tool, Owl-M requires users to have the self-discipline and motivation to use it regularly and complete their study goals.
- **Possible distractions:** While the app is designed to be user-friendly and engaging, it is still a digital tool that may offer distractions or temptations that could take away from the user's study time and focus.

CHAPTER-V APPLICATIONS

- ## Self-study

Students who prefer to study on their own can use Owl-M to create their own study materials, such as flashcards and notes, and test their knowledge with quizzes and assessments. The goal setting and reminders feature can help them stay motivated and on track with their study goals.

- ## Group study

Students who prefer to study with others can use Owl-M to collaborate on study materials, share flashcards and notes, and test each other with quizzes and assessments. The app's gamification feature can make studying more fun and engaging for group study sessions.

- ## Language learning

Owl-M's customizable flashcard system and note-taking feature can be particularly useful for language learners, allowing them to create and organize vocabulary lists and practice their listening and speaking skills with audio and image-based flashcards.

- ## Exam preparation

Owl-M's quiz and assessment features can be used for exam preparation, allowing students to test their knowledge and track their progress over time. The app's analytics and insights feature can also help identify areas of strength and weakness, allowing students to focus their study efforts more effectively.

- ## Professional development

  Owl-M can be used by professionals or individuals seeking to learn new skills, such as coding or graphic design. The app's customizable flashcards and note-taking features can help users organize and retain new information, while the quiz and assessment features can test their knowledge and track their progress.

CHAPTER-VI CONCLUSION

- A Material design study app is a comprehensive study tool that offers various features to help users learn, retain, and recall information more effectively.
- With its customizable flashcards, note-taking system, quizzes, assessments, goal setting and reminder features, Owl-M can be applied in various ways for different types of learners, from self-study to group study, language learning, exam preparation, and professional development.
- While the app does have some limitations, such as limited free features and social capabilities, Owl-M's user-friendly interface and material design make it an enjoyable and engaging tool for students and professionals alike.
- Overall, Owl-M is a valuable study app that can help users achieve their learning goals and succeed academically or professionally.

CHAPTER-VII FUTURE SCOPE

- **Artificial Intelligence Integration**

  The app can integrate artificial intelligence (AI) capabilities to provide personalized learning recommendations, track learning patterns, and offer insights for more effective learning outcomes.

- **Augmented Reality (AR) Integration**

  The app can integrate AR to provide interactive and immersive learning experiences, such as virtual flashcards or quizzes that can be accessed with a smartphone or tablet.

- **Gamification Enhancements**

The app can continue to enhance its gamification features, such as adding leaderboards, badges, and rewards to make learning more fun and engaging**.**

- **Social features Enhancement**

  The app can enhance its social features, such as adding collaboration tools, discussion forums, or the ability to share study materials, to make it easier for students to connect and learn with others.

- **Integration with Learning Management Systems**

  The app can integrate with learning management systems (LMS) used in schools and universities, allowing teachers to monitor student progress, assign tasks, and provide feedback.

- **Localization**

  The app can localize its content and user interface to cater to the specific needs and preferences of different cultures and languages.

- **Audio and video Learning**

The app can expand its features to include more audio and video-based learning materials, such as podcasts or instructional videos, to cater to different learning styles.

CHAPTER-VIII APPENDIX

SOURCECODE:

MAIN ACTIVITY

```
package com.example.owlapplication

import android.content.Context

import android.content.Intent

import android.os.Bundle

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

import androidx.compose.foundation.Image

import androidx.compose.foundation.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,
                )
            }
        }
    }
}
```

USER

```
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?,
)
```

USERDAO

```
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)
}
```

USERDATABASE

```
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

```
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) {
```

```kotlin
companion object {
    private const val DATABASE_VERSION = 1
    private const val DATABASE_NAME = "UserDatabase.db"

    private const val TABLE_NAME = "user_table"
    private const val COLUMN_ID = "id"
    private const val COLUMN_FIRST_NAME = "first_name"
    private const val COLUMN_LAST_NAME = "last_name"
    private const val COLUMN_EMAIL = "email"
    private const val COLUMN_PASSWORD = "password"
}

override fun onCreate(db: SQLiteDatabase?) {
    val createTable = "CREATE TABLE $TABLE_NAME (" +
            "$COLUMN_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "$COLUMN_FIRST_NAME TEXT, " +
            "$COLUMN_LAST_NAME TEXT, " +
            "$COLUMN_EMAIL TEXT, " +
            "$COLUMN_PASSWORD TEXT" +
            ")"

    db?.execSQL(createTable)
}
```

```
    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int)
{
        db?.execSQL("DROP TABLE IF EXISTS $TABLE_NAME")
        onCreate(db)
    }


    fun insertUser(user: User) {
        val db = writableDatabase
        val values = ContentValues()
        values.put(COLUMN_FIRST_NAME, user.firstName)
        values.put(COLUMN_LAST_NAME, user.lastName)
        values.put(COLUMN_EMAIL, user.email)
        values.put(COLUMN_PASSWORD, user.password)
        db.insert(TABLE_NAME, null, values)
        db.close()
    }


    @SuppressLint("Range")
    fun getUserByUsername(username: String): User? {
        val db = readableDatabase
        val cursor: Cursor = db.rawQuery("SELECT * FROM $TABLE_NAME
WHERE $COLUMN_FIRST_NAME = ?", arrayOf(username))
```

```
        var user: User? = null
        if (cursor.moveToFirst()) {
            user = User(
                id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
                firstName =
cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
                lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
                email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
                password =
cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
            )
        }
        cursor.close()
        db.close()
        return user
    }
    @SuppressLint("Range")
    fun getUserById(id: Int): User? {
        val db = readableDatabase
        val cursor: Cursor = db.rawQuery("SELECT * FROM $TABLE_NAME
WHERE $COLUMN_ID = ?", arrayOf(id.toString()))
        var user: User? = null
        if (cursor.moveToFirst()) {
            user = User(
                id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
                firstName =
cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
                lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
                email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
                password =
cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
            )
        }
        cursor.close()
        db.close()
        return user
    }


    @SuppressLint("Range")
    fun getAllUsers(): List<User> {
        val users = mutableListOf<User>()
        val db = readableDatabase
        val cursor: Cursor = db.rawQuery("SELECT * FROM $TABLE_NAME",
null)
        if (cursor.moveToFirst()) {
            do {
                val user = User(
                    id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
                    firstName = cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
                    lastName = cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
                    email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
                    password = cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
                )
                users.add(user)
            } while (cursor.moveToNext())
        }
        cursor.close()
        db.close()
        return users
    }

}
```

LOGINACTIVITY

```
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)
}
```

REGISTERACTIVITY

```
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)
}
```

MAINACTIVITY2

```
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
        )
}
}
```

MAINACTIVTY3

```
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

```
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
        )
}
}
```

MAINACTIVITY5

```
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
        )
}
 }
```