Owl-M: A Material Design Study App

Introduction

1.1 Overview

- Owl-M is a study app that combines elegant design with powerful functionality to help users stay organized and focused while studying. The app follows Google's Material Design principles, which prioritize simplicity, clarity, and visual hierarchy.

- One of the key features of Owl-M is its ability to help users create and manage study schedules. With the app, users can set study goals and reminders, and then create a customized study plan that fits their unique needs and preferences. The app also includes a progress tracker that helps users stay on track and monitor their progress over time.

- In addition to its scheduling features, Owl-M also includes a variety of study tools that can help users learn more effectively. For example, the app includes a flashcard feature that allows users to create digital flashcards to help them memorize important information. The app also includes a quiz feature that lets users test their knowledge and identify areas where they may need to focus more attention.

- Owl-M also includes a note-taking tool that makes it easy for users to jot down important information during lectures or while studying. The app allows users to create and organize notes, and even includes a search function to help users quickly find information they need.

- Overall, Owl-M is a versatile study app that combies design and functionality to help users stay organized, focused, and productive while studying.

1.2 Purpose

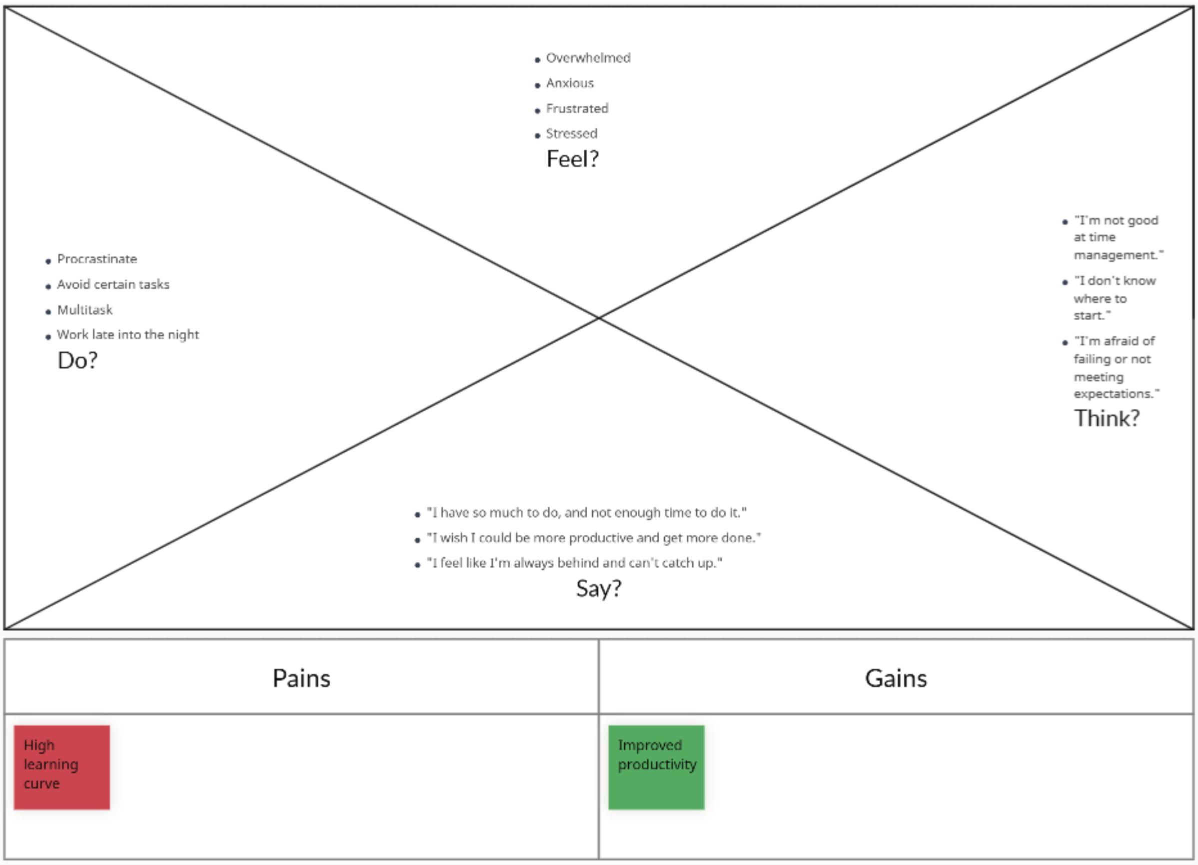
- The purpose of Owl-M is to provide a study app that follows Google's Material Design principles and offers an intuitive and engaging user experience to help students stay organized and productive while studying.

- The app aims to help students improve their study habits and learn more effectively by providing features such as flashcards, quizzes, and note-taking tools. It also allows users to create and manage their own study schedules and set reminders for important deadlines, helping them stay on top of their workload.

- Overall, Owl-M's purpose is to make studying more enjoyable and efficient by providing a sleek and customizable interface, as well as a variety of powerful features to help users stay focused and motivated.

Problem Definition & Design Thinking

2.1 Empathy Map



- Overwhelmed
- Anxious
- Frustrated
- Stressed

Feel?

- Procrastinate
- Avoid certain tasks
- Multitask
- Work late into the night

Do?

- "I'm not good at time management."
- "I don't know where to start."
- "I'm afraid of failing or not meeting expectations."

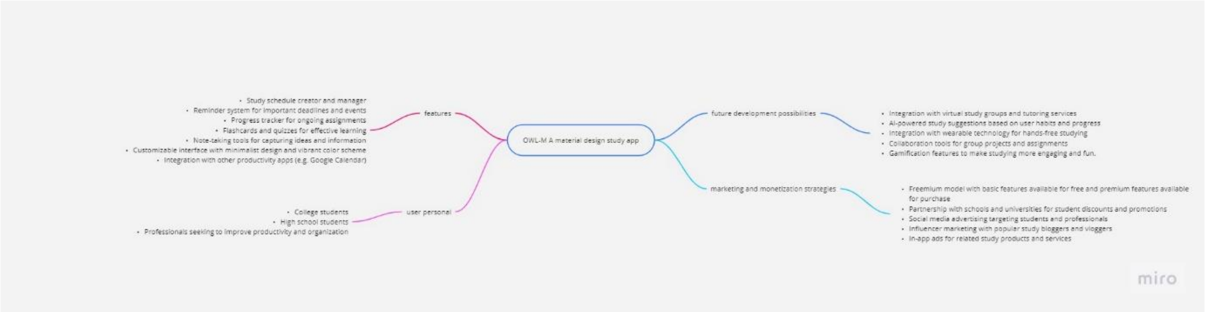
Think?

- "I have so much to do, and not enough time to do it."
- "I wish I could be more productive and get more done."
- "I feel like I'm always behind and can't catch up."

Say?

Pains	Gains
High learning curve	Improved productivity

2.2 Ideation & Brainstroming Map



OWL-M A material design study app

features

- Study schedule creator and manager
- Reminder system for important deadlines and events
- Progress tracker for ongoing assignments
- Flashcards and quizzes for effective learning
- Note-taking tools for capturing ideas and information
- Customizable interface with minimalist design and vibrant color scheme
- Integration with other productivity apps (e.g. Google Calendar)

user personal

- College students
- High school students
- Professionals seeking to improve productivity and organization

future development possibilities

- Integration with virtual study groups and tutoring services
- AI-powered study suggestions based on user habits and progress
- Integration with wearable technology for hands-free studying
- Collaboration tools for group projects and assignments
- Gamification features to make studying more engaging and fun.

marketing and monetization strategies

- Freemium model with basic features available for free and premium features available for purchase
- Partnership with schools and universities for student discounts and promotions
- Social media advertising targeting students and professionals
- Influencer marketing with popular study bloggers and vloggers
- In-app ads for related study products and services

Result

Advantages & Disadvantages

- Advantages: Material Design: Owl-M's user interface is designed following Google's Material Design principles. This means that the app is intuitive, easy to use, and visually appealing, making it a pleasure to use.

- Organization: Owl-M helps users stay organized by providing tools to create and manage study schedules, set reminders, and keep track of progress. With the app, users can easily see what tasks they need to complete and when, reducing the chances of missing important deadlines.

- Learning Tools: Owl-M includes various features to help users learn more effectively, including flashcards, quizzes, and note-taking tools. These tools make it easy for users to review material, test their knowledge, and take notes while studying.

- Customization: Owl-M offers a customizable interface, allowing users to personalize their study experience. Users can choose their preferred color scheme and customize the app's layout to suit their needs and preferences.

- Focus and Concentration: Owl-M's minimalist design and calming color scheme create a distraction-free environment that encourages focus and concentration. This helps users stay on task and avoid getting sidetracked while studying.

Disadvantages:

  - Limited features: While Owl-M offers a range of useful study tools, it may not have all the features that some users need. For example, the app does not offer integrated access to external resources such as textbooks or research materials, which may be important for some students.

  - No collaboration features: Owl-M is designed for individual use and does not include collaboration features, such as the ability to share notes or schedules with others. This may be a drawback for students who prefer to work with study groups or have a more collaborative approach to learning.

  - Cost: While the app offers a free trial period, users will need to pay a monthly or annual subscription fee to continue using it beyond that period. Some students may find this cost prohibitive, especially if they are already paying for other educational resources or materials.

  - Limited availability: As of now, the app is only available for iOS devices, which may limit its accessibility to students who use other operating systems or devices.

Applications

- Owl-M is an application designed to help students improve their study habits and stay organized. The app is available for both Android and iOS devices and can be downloaded from their respective app stores.

- Once downloaded, users can create their own study schedules by setting up study sessions, creating reminders for deadlines, and keeping track of their progress. The app's user-friendly interface allows for easy navigation and customization to cater to individual needs.

- Owl-M also provides a variety of study tools, including flashcards, quizzes, and note-taking features. These tools can help students learn and retain information more effectively. The app's minimalist design and soothing color scheme creates a conducive environment for studying and promotes focus and concentration.

- One of the most notable features of Owl-M is its ability to synchronize data across multiple devices. This means that students can access their study schedules and progress from any device they use. This feature allows for seamless integration of study schedules with daily activities, which ultimately enhances productivity.

- In summary, Owl-M is an innovative application that offers a comprehensive solution to the challenges that students face while studying. The app's intuitive design and powerful features make it an excellent choice for students of all levels who are looking to improve their study habits.

Conclusion

- In conclusion, Owl-M is a beautifully designed study app that provides an intuitive and engaging user experience. The app is based on Google's Material Design principles, which create a sleek and customizable interface that encourages focus and concentration.

- With Owl-M, students and professionals can stay organized, productive, and on top of their game. The app offers a variety of features to help users learn more effectively, including flashcards, quizzes, note-taking tools, and the ability to create and manage study schedules.

- Overall, Owl-M is a great tool for anyone looking to improve their study habits and increase their productivity. Its minimalist design and vibrant color scheme make studying a calming and enjoyable experience, while its powerful features help users stay on track and achieve their goals.

Future Scope

- Integration with learning management systems: Owl-M could integrate with learning management systems used by universities and schools to streamline the student experience. This would allow students to access their course materials, assignments, and grades directly within the app.

- Social learning features: Adding social learning features such as study groups, discussion forums, and peer-to-peer support could enhance the collaborative learning experience for students.

- Personalized learning: Owl-M could use machine learning algorithms to personalize the learning experience for each student based on their individual learning style and progress.

- Gamification: The app could incorporate gamification elements, such as badges, rewards, and leaderboards, to motivate students and make studying more engaging.

- Augmented reality: Owl-M could leverage augmented reality technology to provide interactive and immersive learning experiences, making studying more fun and engaging.

Appendix

A. Source Code

User data class:

```
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 interface :

```
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 class :

```
  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 class :

```
package com.example.owlapplication

import android.annotation.SuppressLint
import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class UserDatabaseHelper(context: Context) :
    SQLiteOpenHelper(context, DATABASE_NAME, null,
DATABASE_VERSION) {

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

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

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

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

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

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

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

}
```

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)
}
```

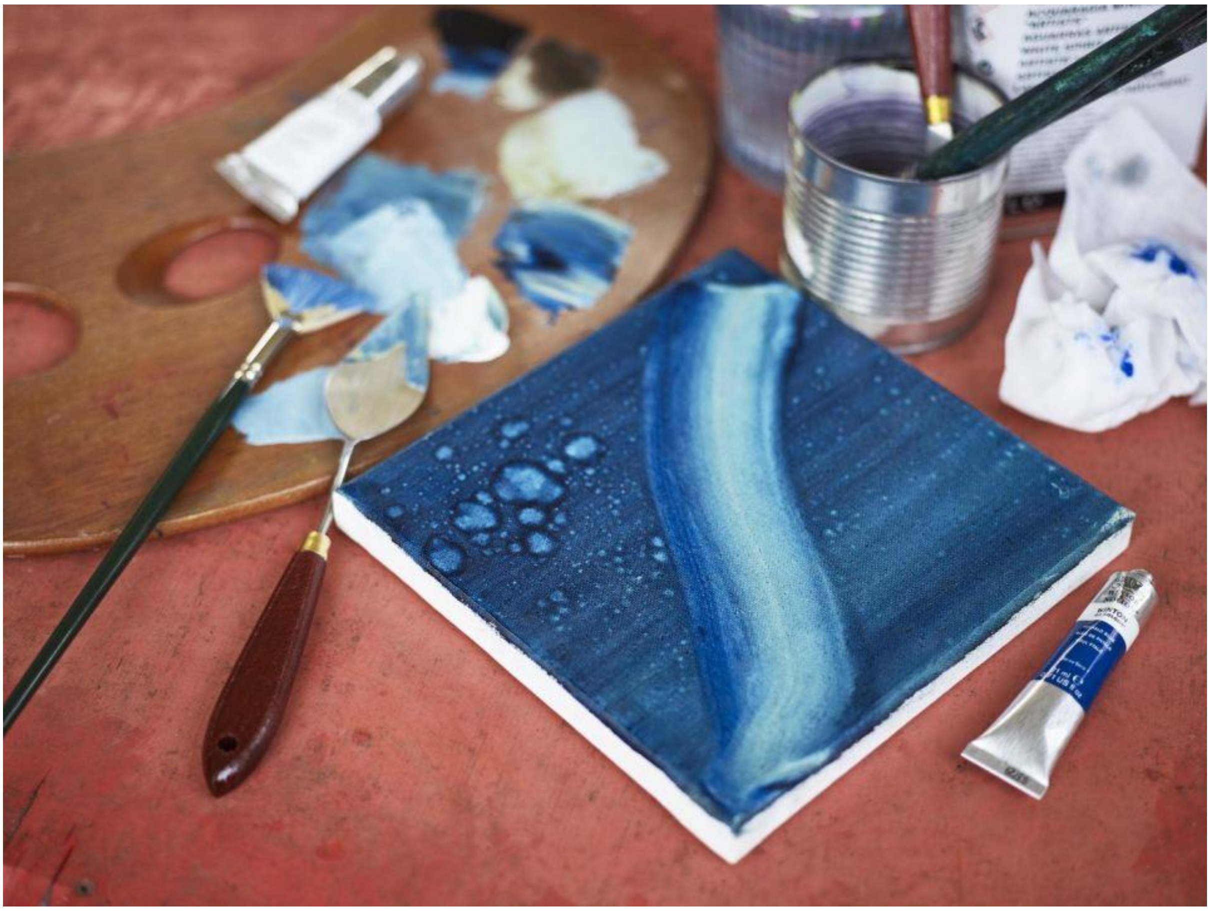
RegisterActivity.dt :

```
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")
```

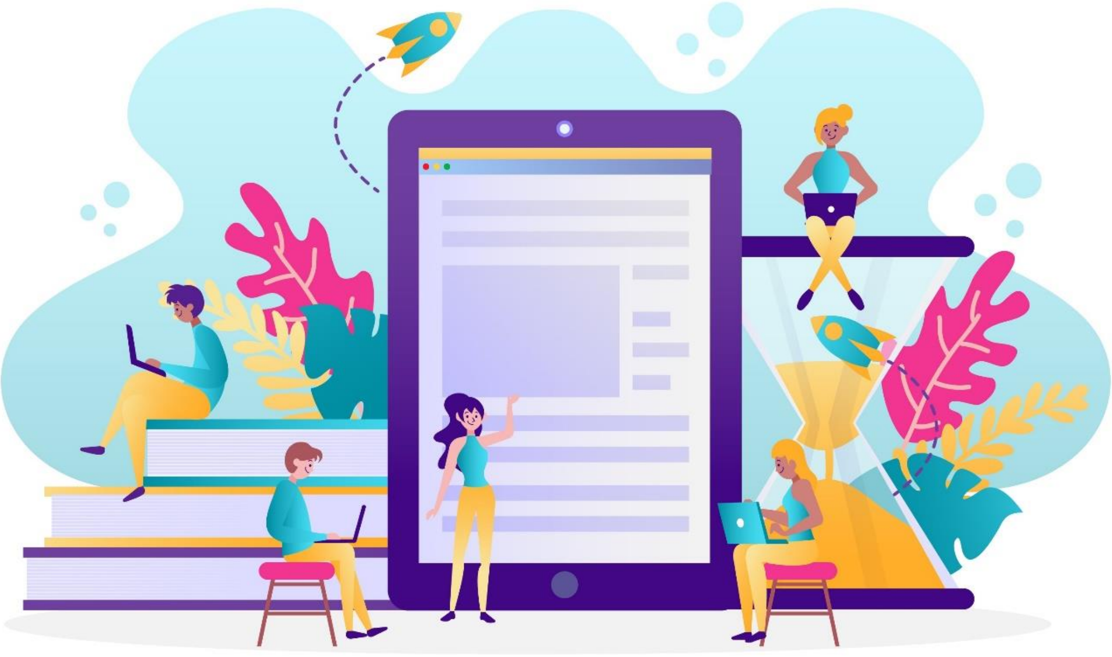
```
            }
        }
    }
}
private fun startLoginActivity(context: Context) {
    val intent = Intent(context, LoginActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
```

MainActivity.dt :

Drawable images:



BRAND
BRANDING
Target
Innovation
Strategy
Marketing
Story
Advertising
Awareness
Quality
Loyalty



Mainactivity 2.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
)
```

```
    }
}
```

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
        )




    }
}
```

Mainactivity 4.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
)
```

```
    }
}
```

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.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
        )




    }
}
```

Androidmanifest .Xml

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        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>
```