CONTENTS

1. INTRODUCTION

1.1 Overview

The Project Study app is a mobile application designed to facilitate the documentation process for students and researchers who are conducting research projects. The app allows users to create and organize their project notes, research data, and findings in a structured and efficient manner**.**

The app also has a collaboration feature that allows multiple users to work on the same project simultaneously. This is particularly useful for team projects where multiple users need to access and contribute to the project data.

The app's search functionality allows users to quickly and easily find the information they need. Users can search by keyword, date, or project category.

1.2 Purpose

The purpose of the Project Study app is to provide a centralized and efficient platform for students and researchers to document and organize their research projects.

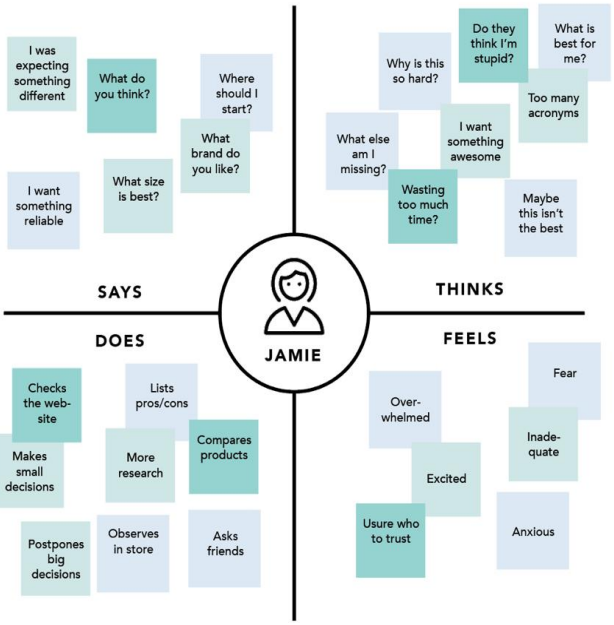
The app aims to simplify the research process and help users stay on track by providing features such as setting deadlines, creating tasks, and setting reminders.

The app's collaboration feature is designed to promote teamwork and facilitate group projects by allowing multiple users to access and contribute to the same project data. This feature can also help users to receive feedback and suggestions from their colleagues, which can improve the quality of their research.

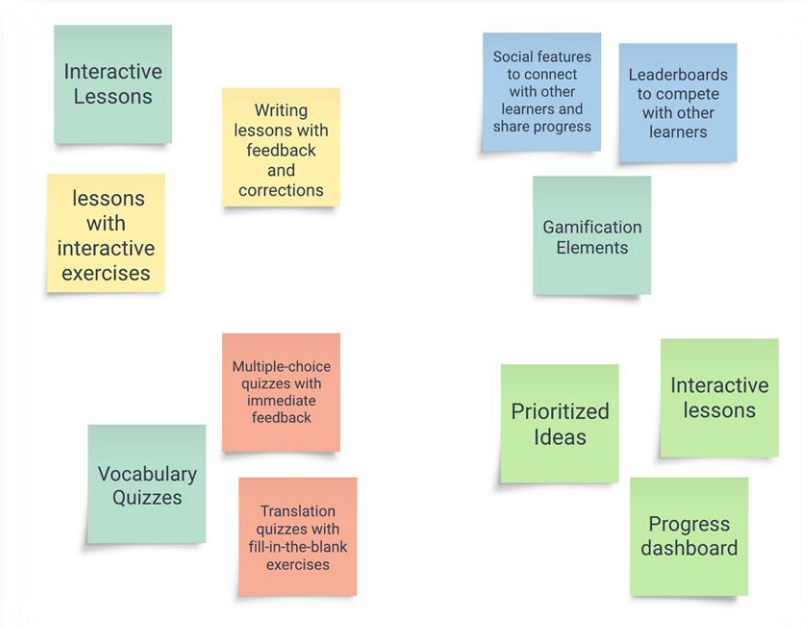
The app's search functionality is another key feature that helps users to quickly and easily find the information they need. This is particularly useful for researchers who may have a large volume of data to sift through.

2. PROBLEM DEFINITION AND DESIGN THINKING

2.1 Empathy Map

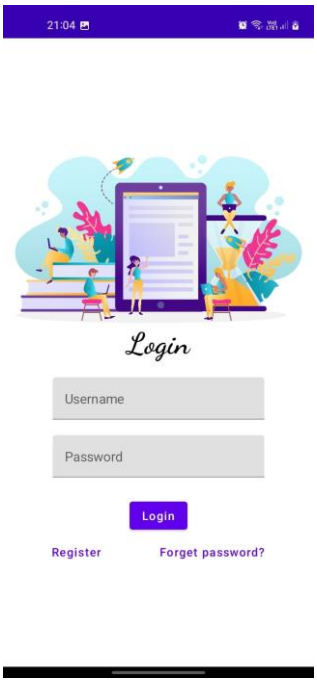


2.2 Ideation and Brainstorming Map

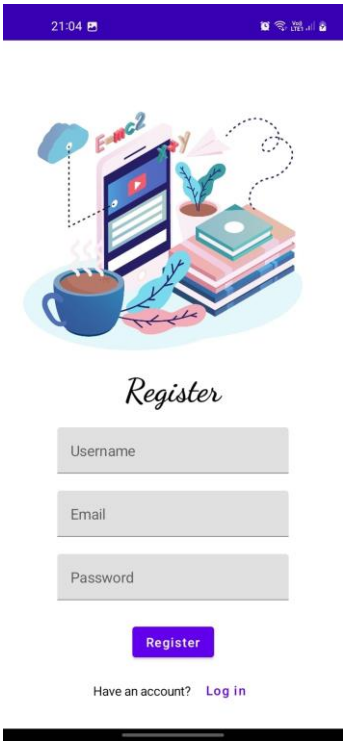


3. RESULT

Login Page:

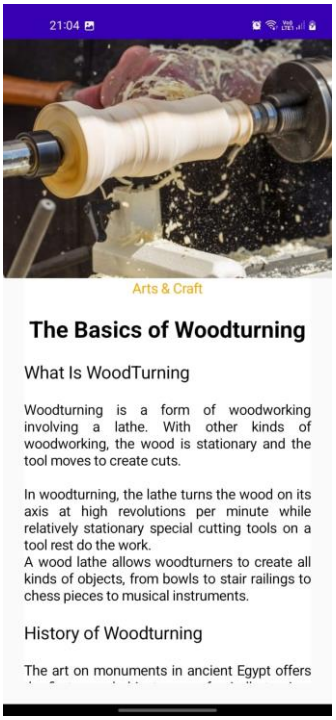


Register Page:



Main Page:

Book Page:



4. ADVANTAGES AND DISADVANTAGES

Advantages of using Study App:

- Convenient Accessibility
- Permanent Source of Education
- Improved Customer Engagement
- Diversified Content Interactive
- Learning Round the Clock
- Availability
- Cost Effectiveness
- Systematic and Improved Learning Ability

Disadvantages of Using Study App:

- Unexpected Software and Hardware issues
- No Physical Interaction
- Restricted Feedback
- No Direct Response
- Lack of Technology
- Knowledge Leads to Disparity among Children

5. APPLICATION

A study app can be applied in various areas and contexts where learning and education are involved. Some examples include:

- Schools and universities: Study apps can be used in traditional educational settings to supplement classroom learning and provide students with additional resources and materials.
- Distance learning: Study apps can be used in online or remote learning contexts to provide students with interactive lessons, quizzes, and other resources that they can access from anywhere.
- Continuing education: Study apps can be used in professional development and continuing education contexts to provide employees with ongoing training and learning opportunities.

6. CONCLUSION

The development of an educational study app can provide a valuable tool for students of all ages and in various educational settings.

The purpose of the study app is to provide students with an interactive and engaging platform for learning, incorporating features such as interactive lessons, vocabulary quizzes, progress tracking, audio and video resources, and gamification elements.

The creation of a brainstorming map helped generate ideas and prioritize key features for the study app, which can be applied in various areas, including schools

and universities, distance learning, continuing education, test preparation, language learning, personal development, and skill-based training.

7. FUTURE SCOPE

The future scope for the educational study app project is vast and varied. Some potential areas for further development and expansion include:

- Artificial Intelligence: Integrating AI technology into the study app can personalize the learning experience for individual users by providing tailored lessons, quizzes, and resources based on their learning style, preferences, and progress.
- Augmented Reality and Virtual Reality: Incorporating AR and VR technology into the study app can provide immersive and interactive learning experiences that can help students better understand complex concepts, particularly in STEM fields.
- Collaborative Learning: Adding features that enable users to collaborate and communicate with other students or teachers within the app can enhance the learning experience by fostering discussion and peer-to-peer learning.

8. APPENDIX

A. Source code:

```
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)
}
```