ANDROID APPLICATION DEVELOPMENT

A MATERIAL DESIGN STUDY APP

TEAM ID: NM2024TMID05614

Submitted By

SHREE DHARSHINI M(Team leader)- AB883A283C7558988D049EAD17F03882

LAKSHANA S(Team Member)- 43806CE4C0DD447B8474B6BE349FD6CD

SANDHIYA S (Team Member)- 958FE1F95ED05DABA3F4ED92DF345324

SARIKA M(Team Member) - FCF580164D0F58AA0054D861DD7EC9BB

SEMESTER-V

B.E COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR-2024-2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY REGIONAL CAMPUS COIMBATORE

COIMBATORE-641046

NOVEMBER 2024

TABLE OF CONTENT:

1.ABSTRACT

The Owl Study Material App is an innovative mobile application designed to enhance the learning experience for students across various educational levels. By providing a comprehensive library of study resources, including textbooks, notes, practice quizzes, and video tutorials, the app offers a personalized, user-friendly platform to help students improve their academic performance. Features such as customizable study plans, progress tracking, and a vast database of learning materials make the app an indispensable tool for efficient study. With a focus on ease of use and accessibility, the Owl Study Material App aims to bridge the gap between traditional learning methods and modern technology, ensuring that students have the resources they need to succeed at their fingertips.

In today's fast-paced educational environment, access to organized and readily available study materials is essential for effective learning. The "Study Material App" aims to provide a comprehensive platform where students can access, organize, and manage study resources in a user-friendly and efficient manner. This app will cater to a wide range of academic levels and subjects, enabling students to find relevant notes, textbooks, sample papers, and multimedia content, all within a single platform. Key features include categorized resources, search functionality, personalized content recommendations, and offline access. The app is designed with an intuitive user interface and integrates interactive tools, such as flashcards and quizzes, to enhance user engagement and knowledge retention. Through this app, students will benefit from streamlined study experiences that support academic success and accessibility to quality resources. This project will significantly impact students' learning efficiency, fostering a more productive study environment.

2.TOOLS AND VERSIONS:

To ensure optimal performance and seamless user

experience, the following are the system requirements for running

Study material app.

Android

• **Operating System**: Android 7.0 (Nougat) or higher

Processor: Quad-core 1.4 GHz or higher

• **RAM**: 2 GB or more

• **Storage**: 100 MB of free storage for the app and cache

• **Internet**: Wi-Fi or mobile data (4G/5G recommended for optimal performance)

• **Other Requirements**: Google Play Services (for push notifications and app updates)

- Bluetooth (for voice and file sharing in close proximity)
- GPS (optional, for location-based features)**iOS**
- **Operating System**: iOS 11.0 or higher

2. Web Application (Browser-Based)

• **Operating System**: Windows, macOS, or Linux (any modern OS with web browser support)

• **Web Browser**: L• atest version of Chrome

• **Processor**: Dual-core 2.0 GHz or higher

• **RAM**: 4 GB or more

3. Desktop Application (Windows/macOS/Linux)

Windows

• **Operating System**: Windows 10 or higher (64-bit recommended)

• **Processor**: Intel i3 or AMD equivalent (dual-core 1.8 GHz or higher)

• **RAM**: 4 GB or more

• **Storage**: 200 MB of free disk space

• **Internet**: Stable broadband internet connection (4G or higher recommended)

3.SAMPLE CODE:

#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.*
me.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
                        )
                    )
                error = "Please fill all fields"
    }
}
private fun startMainPage(context: Context) {
    val intent = Intent(context, MainActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
```

#MainActivity.kt:

```
package com.example.owlapplication

import android.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
            Greeting()
        }
    }
}
        Spacer(modifier = Modifier.height(60.dp))



        )

),
            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.s
    }
```

#build.gradle

```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
}

android {
    namespace 'com.example.owlapplication'
    compileSdk 33

    defaultConfig {
        applicationId "com.example.owlapplication"
        minSdk 24
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
        vectorDrawables {
            useSupportLibrary true
        }

    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    androidTestImplementation "androidx.compose.ui:ui-test-junit4:$compose_ui_version"
    debugImplementation "androidx.compose.ui:ui-tooling:$compose_ui_version"
    debugImplementation "androidx.compose.ui:ui-test-manifest:$compose_ui_version"
}
    }
```

```
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
```

#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 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
            )
        )
    })
```

#Mainactivity.kt2

```
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
)
```

#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
            }
        }
    }
```

}

4.TESTING:

1.In registeration process validation is given to the mail and pass word text box.

i)"@" symbol must be inserted in the mail texg box.

ii)atleast 4 characters required lesser than 4 characters will not be accepted

5.PROJECT HURDLES:

- Ensuring compatibility with different Android versions and devices.
- Slow load times and lag due to a large volume of study materials or complex UI elements.
- Keeping third-party libraries up-to-date without breaking your app's functionality.
- Users may not have reliable internet access all the time, which could disrupt app usage.
- Frequent network requests can cause delays, especially over slow or unreliable connections.

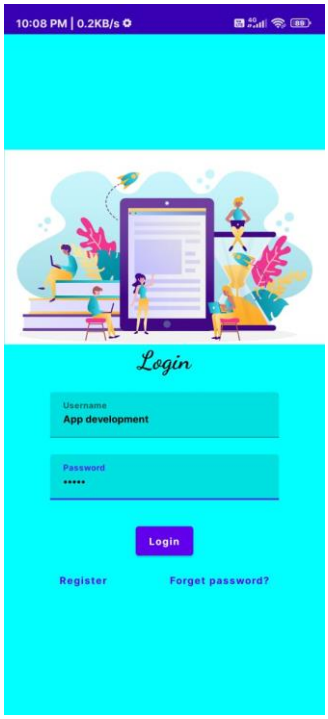
6.OUTPUT AND DEMO LINK:

Drive link:

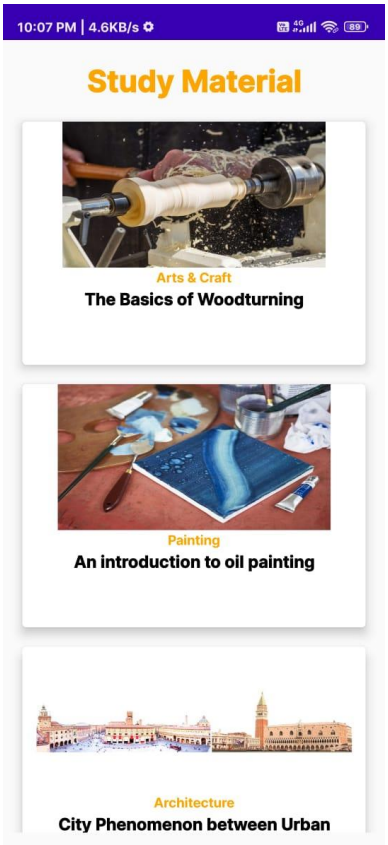
https://drive.google.com/file/d/1kUdUbUldR0Y1K7QZkDF8xoF5tP_9kfX6

STEP1:REGISTER

STEP2:LOGIN



STEP3:



STEP4:

7.CONCLUSION:

To sum up, this study on Android app development gives a clear picture of what goes into building apps for Android and the challenges faced along the way:

1. Wide Reach: Android is popular worldwide, making it a great choice for reaching large and diverse groups of people. Developers can create many types of apps, from games and tools to health and learning apps.
2. Development Steps: Building an Android app involves several steps: planning, designing, coding, testing, and launching. Java and Kotlin are commonly used, and each language has its own benefits for speed and performance.
3. User-Friendly Design: A good app is easy to use and visually appealing. Focusing on a smooth, attractive design helps keep users engaged and happy.
4. Device Differences: Android apps need to work on many types of devices, with different screen sizes and versions. Making sure an app runs well on all these devices can be challenging.
5. Security: Protecting users' personal information is crucial. Developers must take strong security steps to keep data safe.
6. New Trends: Android is always improving, with new features like AI, virtual reality, and support for smart devices. Following these trends helps developers create exciting, modern apps.

In short, this study shows that Android app development is complex but rewarding. Developers who stay updated and focused on quality and security can create successful apps that users enjoy.