



# Personal Mu

Presented BY: Penpicha 6380667, Proifon 6380144, Sittipoj 6380361



# About us

Software Design  
Software Engineer  
Mobile Application Programming



# APPLICATION





**SIGN IN**

- USERNAME -

- PASSWORD -

**SIGN IN**

**CANCEL**

**SIGNUP**

**SIGN IN PAGE**



**SIGN UP**

- USERNAME -

- PASSWORD -

**SIGN UP**

**SIGN UP PAGE**

# **SIGN IN AND SIGN UP PAGE**

## WELCOME

Please answer these questions

What color is your vein on the wrist?

- Green       Purple

Skin color after being exposed to the sun?

- Dark Skin       Reddish Skin

Does gold or Silver jewellery suit you the best?

- Better with gold       Better with silver

**SUBMIT**

# Personal Color test questions



Your personal color is ...

COOL TONE

- *Return to SignIn Page* -

# RESULT PAGE

☰ PersonalMu

TODAY



12 JULY  
WEDNESDAY

WARM TONE

LUCKY COLOR:

Work: 

Money: 

Love: 

Luck: 

Misery: 

# Homepage && Menu tab

Today's Color

Lucky Draw

Check tone

12 JULY  
EDNESDAY

☰ PersonalMu

TODAY

12 JULY  
WEDNESDAY



WARM TONE

LUCKY COLOR:

Work: 

Money: 

Love: 

Luck: 

Misery: 

☰ PersonalMu

TODAY

13 JULY  
THURSDAY



COOL TONE

LUCKY COLOR:

Work: 

Money: 

Love: 

Luck: 

Misery: 

# Daily Lucky Color

## FORTUNE TELLER

Shake for the fortune result



## LUCKY DRAW

## FORTUNE TELLER

Shake for the fortune result

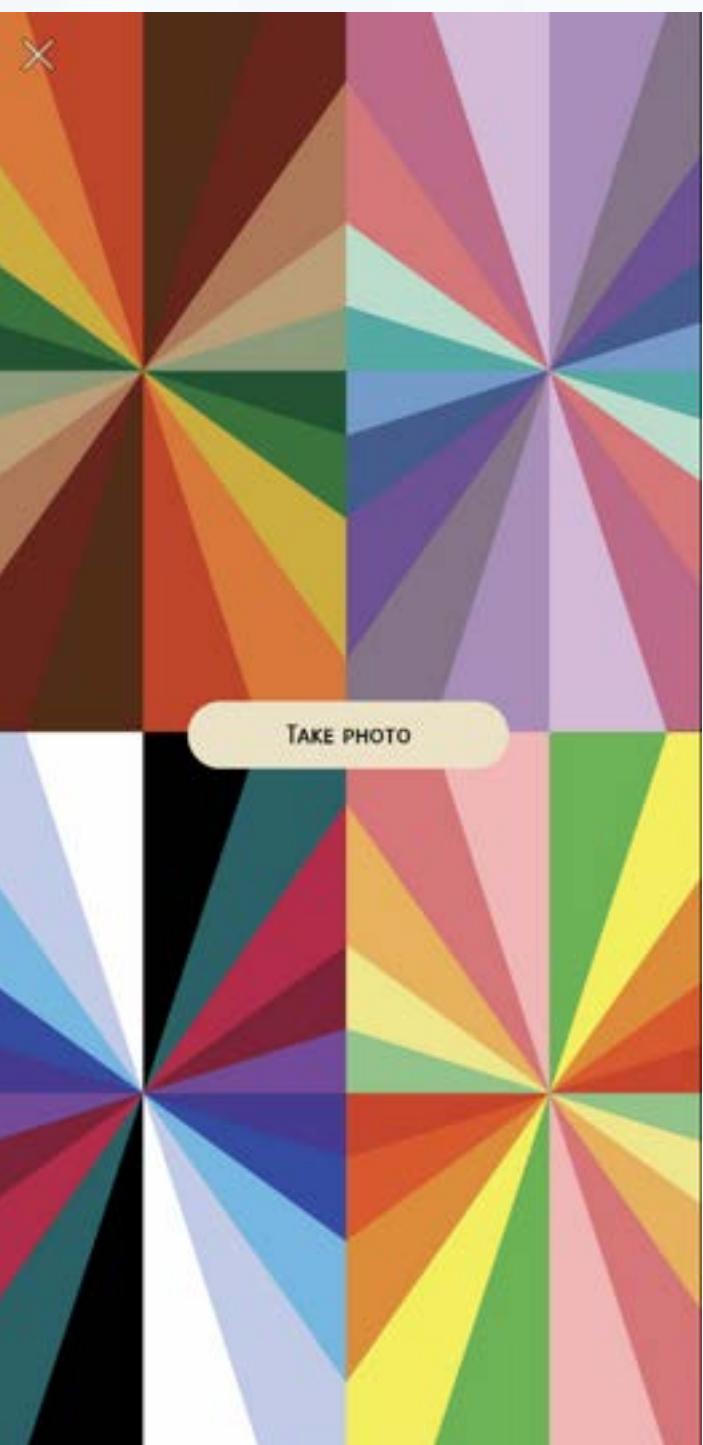


Do not lose yourself in the past. Do not lose yourself in the future. Do not get caught in your anger, worries, or fears. Come back to the present moment, and touch life deeply.

This is mindfulness.



## Personal Color Camera



# Algorithms

## Sensors

- Camera
- Accelerometer

## Json file

Fortune Teller

## Database

- Firebase
- User.txt file

```
class SignUpActivity : AppCompatActivity() {  
  
    //private val file = "users.txt"  
    //lateinit var usrList: MutableList<User>  
    lateinit var userSignText: EditText  
    lateinit var passSignText: EditText  
  
    /* Penicha +1 */  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_sign_up)  
  
        val addBtn = findViewById<Button>(R.id.addBtn)  
        userSignText = findViewById(R.id.editText)  
        passSignText = findViewById(R.id.editText2)  
  
        addBtn.setOnClickListener { it: View ->  
            val intent = Intent(packageContext, Question::class.java)  
            intent.putExtra(name: "user", userSignText.text.toString())  
            intent.putExtra(name: "pw", passSignText.text.toString())  
            startActivity(intent)  
  
            Log.d(tag: "user", userSignText.text.toString())  
            Log.d(tag: "upass", passSignText.text.toString())  
        }  
    }  
}
```

## Sign Up Activity

```
radioGroup3.setOnCheckedChangeListener { group, checkedId ->  
    when(checkedId){  
        R.id.gold ->{warm += 1}  
        R.id.silver ->{cool +=1}  
    }  
}  
  
submitBtn.setOnClickListener { it: View ->  
    if (warm > cool){  
        tone = "Warm tone"  
    } else { tone = "Cool tone"}  
    user_name = intent.getStringExtra(name: "user").toString()  
    password = intent.getStringExtra(name: "pw").toString()  
    saveData(user_name, password, tone)  
    Log.d(tag: "warm tone", tone)  
    val intent = Intent(packageContext: this, PersonalColor::class.java)  
    intent.putExtra(name: "Tone", tone)  
    startActivity(intent)  
}  
/* Beamtaab */  
private fun saveData(username: String, passw: String, skintone: String) {  
  
    val userData = User(username = username, password = passw, tone= skintone)  
    val line = userData.username + "," + userData.password + "," + userData.tone  
    Log.d(tag: "Quser1", user_name)  
    Log.d(tag: "Quser1", passw)  
    Log.d(tag: "Quser1", tone)  
}
```

## Question

```
package com.egci428.personalmu

import ...

// PersonalColor +1
class PersonalColor : AppCompatActivity() {
    // PersonalColor +1
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_personal)
        var tone = intent.getStringExtra( name: "Tone")
        var tv = findViewById<TextView>(R.id.personalColor)
        tv.text = tone

        val returnBtn = findViewById<Button>(R.id.returnBtn)

        returnBtn.setOnClickListener { it: View
            // saveData()
            val intent = Intent( packageContext: this, MainActivity::class.java)
            startActivity(intent)
        }
    }
}
```

## Result

```
signUpBtn.setOnClickListener { it: View!
    val intent = Intent( packageContext: this, SignUpActivity::class.java)
    startActivity(intent)
}

signInBtn.setOnClickListener { it: View!
    uname = userText.text.toString()
    pname = passText.text.toString()
    if (!uname.isNullOrEmpty() && !pname.isNullOrEmpty()) {
        var loginSuccessful = false // Flag to track login success
        try {
            val fin = openFileInput(file)
            val mfile = InputStreamReader(fin)
            val br = BufferedReader(mfile)
            var line = br.readLine()
            while (line != null) {
                val userItem = line.split( ...delimiters: ", ")
                if (userItem[0] == uname & userItem[1] == pname) {
                    loginSuccessful = true // Set the flag to true
                    val dataToSend = if (line.contains( other: "Warm")) 0 else 1
                    Toast.makeText(applicationContext, text: "Login Success")
                    val intent = Intent( packageContext: this, Drawer::class.java)
                    intent.putExtra( name: "dataToSend", dataToSend)
                    startActivity(intent)
                    break // Exit the loop when login is successful
                }
                line = br.readLine()
            }
        } catch (e: IOException) {
            e.printStackTrace()
        }
    }
}
```

## Sign In Activity

```
drawerLayout = findViewById<DrawerLayout>(R.id.drawerLayout)

val toolbar = findViewById<Toolbar>(R.id.toolbar)
setSupportActionBar(toolbar)

val navigationView = findViewById<NavigationView>(R.id.nav_view)
navigationView.setNavigationItemSelectedListener(this)

val toggle = ActionBarDrawerToggle( activity, this, drawerLayout, toolbar,
drawerLayout.addDrawerListener(toggle)
toggle.syncState()

if (savedInstanceState == null) {
    navigationView.setCheckedItem(R.id.nav_Colar)
}

val formatter = DateTimeFormatter.ofPattern( pattern: "dd MMMM EEEE")
val current = LocalDateTime.now().format(formatter)

val date = findViewById<TextView>(R.id.DateTV)
date.text = current

FirebaseApp.initializeApp(this)
val db: FirebaseFirestore = FirebaseFirestore.getInstance()

// Load data from Firestore
```

## Home Page

```
fun fetchJson(){

    val textView = findViewById<TextView>(R.id.textView)
    val jsonURL: String = "https://raw.githubusercontent.com/Proifton/Prediction/master/predictions.json"
    val request = Request.Builder()
        .url(jsonURL)
        .build()

    client.newCall(request).enqueue(object : Callback {
        override fun onFailure(call: Call, e: IOException) {
            e.printStackTrace()
        }

        override fun onResponse(call: Call, response: Response) {
            response.use { it: Response ->
                if (!response.isSuccessful) throw IOException("Unexpected code $response")
                for ((name, value) in response.headers) {
                    println("$name: $value")
                }
                val body = response.body!!.string()
                if (body == null) return@use
                val gson = GsonBuilder().create()
                val quotes = gson.fromJson(body, Array<predictions>::class.java)
                textView.text = quotes[0].text
            }
        }
    })
}
```

## Lucky Draw

```
 1 Penpicha
 2
 3 fun takePhoto(view: View){
 4     requestCameraPermission.launch(Manifest.permission.CAMERA)
 5 }
 6
 7 private val requestCameraPermission = registerForActivityResult(ActivityResultContracts.RequestPermission()) {
 8     isSuccess : Boolean ->
 9         if(isSuccess){
10             Log.d( tag: "Take Picture", msg: "Permission granted")
11             takePicture.launch( input: null)
12         } else {
13             Toast.makeText(applicationContext, text: "Camera has no permission")
14         }
15 }
16
17 private val takePicture = registerForActivityResult(ActivityResultContract<Image, Image>()) {
18     bitmap: Bitmap? ->
19         Log.d( tag: "Take Picture", msg: "Show bitmap picture")
20         imageView1.setImageBitmap(bitmap)
21         imageView2.setImageBitmap(bitmap)
22         imageView3.setImageBitmap(bitmap)
23         imageView4.setImageBitmap(bitmap)
24         textView1.setText("Autumn")
25         textView2.setText("Summer")
26         textView3.setText("Winter")
27         textView4.setText("Spring")
28 }
```

## Check Tone

# DEMO





Thank  
you