

[2CEIT5PE18: MOBILE APPLICATION DEVELOPMENT]

Assignment

AIM- Student Reminder App

Submitted By: KANSAGARA KRISH KALPESHBHAI
Enrollment number: 23012011026



**Ganpat
University**

॥ विद्या समाजोत्कर्षः ॥

**U.V. Patel
College of
Engineering**

**Department of Computer
Engineering/Information Technology**

Student Reminder App

MainActivity.kt

```
package com.example.mad_23012011026

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.content.ContextCompat
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {

    private lateinit var emailEditText: EditText
    private lateinit var passwordEditText: EditText
    private lateinit var loginButton: Button
    private lateinit var signupText: TextView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Enable edge-to-edge
        SystemBarsHelper.enableEdgeToEdge(this)

        setContentView(R.layout.activity_main)

        emailEditText = findViewById(R.id.emailEditText)
        passwordEditText = findViewById(R.id.passwordEditText)
        loginButton = findViewById(R.id.loginButton)
        signupText = findViewById(R.id.signupText)

        setupWindowInsets()
        applyTheme()

        loginButton.setOnClickListener {
            val email = emailEditText.text.toString().trim()
            val password = passwordEditText.text.toString().trim()
        }
    }
}
```

Student Reminder App

```
        if (email.isEmpty() || password.isEmpty()) {
            Toast.makeText(this, "Please enter email and password",
Toast.LENGTH_SHORT).show()
        } else if (email == "student@2025" && password == "123") {
            val intent = Intent(this, HomeActivity::class.java)
            startActivity(intent)
        } else {
            Toast.makeText(this, "Invalid email or password",
Toast.LENGTH_SHORT).show()
        }
    }

    signupText.setOnClickListener {
        val intent = Intent(this, RegisterActivity::class.java)
        startActivity(intent)
    }
}

private fun setupWindowInsets() {
    val mainLayout =
findViewById<androidx.constraintlayout.widget.ConstraintLayout>(R.id.mainLayout)
    val paddingDp = (24 * resources.displayMetrics.density).toInt()

    ViewCompat.setOnApplyWindowInsetsListener(mainLayout) { view, insets ->
        val systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
        val imeInsets = insets.getInsets(WindowInsetsCompat.Type.ime())

        view.setPadding(
            systemBars.left + paddingDp,
            systemBars.top + paddingDp,
            systemBars.right + paddingDp,
            maxOf(systemBars.bottom, imeInsets.bottom) + paddingDp
        )

        insets
    }
}

private fun applyTheme() {
```

Student Reminder App

```
val isDarkMode = DarkModeManager.isDarkMode(this)

val backgroundColor = if (isDarkMode)
    ContextCompat.getColor(this, R.color.darkBackgroundColor)
else
    ContextCompat.getColor(this, R.color.white)

findViewById<androidx.constraintlayout.widget.ConstraintLayout>(R.id.mainLayout).
setBackgroundColor(backgroundColor)

val textColor = if (isDarkMode)
    ContextCompat.getColor(this, R.color.darkEditTextTextColor)
else
    ContextCompat.getColor(this, R.color.editTextColor)

val hintColor = if (isDarkMode)
    ContextCompat.getColor(this, R.color.darkEditTextHintColor)
else
    ContextCompat.getColor(this, R.color.editTextHintColor)

emailEditText.setTextColor(textColor)
emailEditText.setHintTextColor(hintColor)
passwordEditText.setTextColor(textColor)
passwordEditText.setHintTextColor(hintColor)

val editTextBackground = if (isDarkMode)
    ContextCompat.getDrawable(this, R.drawable.edit_text_bg_dark)
else
    ContextCompat.getDrawable(this, R.drawable.edit_text_bg)

emailEditText.background = editTextBackground
passwordEditText.background = editTextBackground
}

}
```

HomeActivity.kt

```
package com.example.mad_23012011026
import android.content.Intent
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.Button
```

KANSAGARA_KRISH_23012011026

Student Reminder App

```
import android.widget.LinearLayout
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import androidx.core.content.ContextCompat
import androidx.drawerlayout.widget.DrawerLayout
import com.google.android.material.appbar.MaterialToolbar
import com.google.android.material.bottomnavigation.BottomNavigationView
import com.google.android.material.navigation.NavigationView

class HomeActivity : AppCompatActivity() {

    private lateinit var greetingText: TextView
    private lateinit var addReminderButton: Button
    private lateinit var stopNotificationButton: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Enable edge-to-edge
        SystemBarsHelper.enableEdgeToEdge(this)

        setContentView(R.layout.activity_home)

        setupWindowInsets()

        greetingText = findViewById(R.id.greetingText)
        addReminderButton = findViewById(R.id.addReminderButton)
        stopNotificationButton = findViewById(R.id.stopNotificationButton)

        // Top App Bar
        val toolbar = findViewById<MaterialToolbar>(R.id.topAppBar)
        setSupportActionBar(toolbar)

        val drawerLayout = findViewById<DrawerLayout>(R.id.drawerLayout)
        val navigationView =
            findViewById<NavigationView>(R.id.navigationView)

        toolbar.setNavigationOnClickListener {
            if
                (drawerLayout.isDrawerOpen(ContextCompat.START)) {

            drawerLayout.closeDrawer(ContextCompat.START)
            } else {

            drawerLayout.openDrawer(ContextCompat.START)
            }
        }
    }
}
```

Student Reminder App

```
navigationView.setNavigationItemSelectedListener { item ->
    when (item.itemId) {
        R.id.nav_home -> {
            // Stay on home
        }
        R.id.nav_add -> {
            startActivity(Intent(this,
AddReminderActivity::class.java))
        }
        R.id.nav_drawer -> {
            drawerLayout.closeDrawer(androidx.core.view.GravityCompat.START)
            true
        }
        R.id.nav_settings -> {
            startActivity(Intent(this, SettingsActivity::class.java))
        }
        R.id.nav_about -> {
            startActivity(Intent(this, AboutActivity::class.java))
        }
        R.id.nav_bottom -> {
            drawerLayout.closeDrawer(androidx.core.view.GravityCompat.START)
            true
        }
        else -> false
    }
}

// Bottom Navigation
val bottomNav = findViewById<BottomNavigationView>(R.id.bottomNav)
bottomNav.setOnItemSelectedListener { item ->
    when (item.itemId) {
        R.id.nav_home -> {
            true
        }
        R.id.nav_add -> {
            startActivity(Intent(this,
AddReminderActivity::class.java))
            true
        }
        R.id.nav_settings -> {
            startActivity(Intent(this, SettingsActivity::class.java))
            true
        }
    }
}
```

Student Reminder App

```
        }
    else -> false
}
bottomNav.selectedItemId = R.id.nav_home

greetingText.text = "Hello, Student ✋"

applyTheme()
loadReminders()

addReminderButton.setOnClickListener {
    val intent = Intent(this, AddReminderActivity::class.java)
    startActivity(intent)
}

// Dark mode is toggled via app bar action or Settings

stopNotificationButton.setOnClickListener {
    NotificationManager.stopNotification(this)
    stopNotificationButton.visibility = android.view.View.GONE
}
}

private fun setupWindowInsets() {
    val drawerLayout = findViewById<DrawerLayout>(R.id.drawerLayout)
    val bottomNav = findViewById<BottomNavigationView>(R.id.bottomNav)
    val toolbar = findViewById<MaterialToolbar>(R.id.topAppBar)

    SystemBarsHelper.setupHomeActivityInsets(drawerLayout, toolbar,
bottomNav)
}

override fun onCreateOptionsMenu(menu: Menu): Boolean {
    menuInflater.inflate(R.menu.top_app_bar_menu, menu)
    return true
}

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
        R.id.action_search -> {
            android.widget.Toast.makeText(this, "Search coming soon",
android.widget.Toast.LENGTH_SHORT).show()
            true
        }
        R.id.action_dark_mode -> {
            DarkModeManager.toggleDarkMode(this)
        }
    }
}
```

Student Reminder App

```
        recreate()
        true
    }
    else -> super.onOptionsItemSelected(item)
}
}

override fun onResume() {
    super.onResume()
    loadReminders()
    checkForNotifications()
}

private fun loadReminders() {
    val reminderContainer =
findViewById<LinearLayout>(R.id.reminderContainer)
    reminderContainer.removeAllViews()

    val sharedPrefs = getSharedPreferences("reminders", MODE_PRIVATE)
    val reminders = sharedPrefs.getStringSet("reminderList", null)

    if (reminders.isNullOrEmpty()) {
        val noReminderText = TextView(this).apply {
            text = "You have no reminders today."
            setTextColor(if
(DarkModeManager.isDarkMode(this@HomeActivity))
                ContextCompat.getColor(this@HomeActivity,
R.color.darkSecondaryTextColor)
            else
                ContextCompat.getColor(this@HomeActivity,
android.R.color.darker_gray))
            textSize = 16f
            textAlign = TextView.TEXT_ALIGNMENT_CENTER
        }
        reminderContainer.addView(noReminderText)
    } else {
        for (reminder in reminders) {
            val parts = reminder.split("\n")
            val title = parts.getOrNull(0) ?: "Title"
            val date = parts.getOrNull(1) ?: "Date/Time"
            val category = parts.getOrNull(2) ?: "Category"

            val rowLayout = LinearLayout(this).apply {
                orientation = LinearLayout.HORIZONTAL
                setPadding(8, 8, 8, 8)
            }
        }
    }
}
```

Student Reminder App

```
    val rowText = TextView(this).apply {
        text = "$title | $dateTime | $category"
        textSize = 14f
        setTextColor(if
(DarkModeManager.isDarkMode(this@HomeActivity))
            ContextCompat.getColor(this@HomeActivity,
R.color.darkPrimaryTextColor)
        else
            ContextCompat.getColor(this@HomeActivity,
android.R.color.black))
    }

    rowLayout.addView(rowText)
    reminderContainer.addView(rowLayout)
}
}
}

private fun applyTheme() {
    val isDarkMode = DarkModeManager.isDarkMode(this)

    // App bar reflects theme; no per-button label needed

    val backgroundColor = if (isDarkMode)
        ContextCompat.getColor(this, R.color.darkBackgroundColor)
    else
        ContextCompat.getColor(this, R.color.white)

    findViewById<androidx.constraintlayout.widget.ConstraintLayout>(R.id.mainLayout).setBackgroundColor(backgroundColor)

    greetingText.setTextColor(if (isDarkMode)
        ContextCompat.getColor(this, R.color.darkPrimaryTextColor)
    else
        ContextCompat.getColor(this, R.color.primaryTextColor))
}

private fun checkForNotifications() {
    val sharedPrefs = getSharedPreferences("notification_status",
MODE_PRIVATE)
    val isNotificationPlaying = sharedPrefs.getBoolean("is_playing",
false)

    if (!isNotificationPlaying) {
        NotificationManager.checkReminders(this)
    }
}
```

Student Reminder App

```
        if (isNotificationPlaying) {
            stopNotificationButton.visibility = android.view.View.VISIBLE
        } else {
            stopNotificationButton.visibility = android.view.View.GONE
        }
    }
}
```

AboutActivity.Kt

```
package com.example.mad_23012011026

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.google.android.material.appbar.MaterialToolbar

class AboutActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Enable edge-to-edge
        SystemBarsHelper.enableEdgeToEdge(this)

        setContentView(R.layout.activity_about)

        setupWindowInsets()

        val toolbar = findViewById<MaterialToolbar>(R.id.aboutToolbar)
        setSupportActionBar(toolbar)
        supportActionBar?.setDisplayHomeAsUpEnabled(true)
        toolbar.setNavigationOnClickListener { finish() }
    }

    private fun setupWindowInsets() {
        val toolbar = findViewById<MaterialToolbar>(R.id.aboutToolbar)
        val rootView =
            findViewById<android.view.ViewGroup>(R.id.aboutMainLayout)

        SystemBarsHelper.setupSimpleActivityInsets(rootView, toolbar)
    }
}
```

Student Reminder App

AddReminderActivity.Kt

```
package com.example.mad_23012011026

import android.app.DatePickerDialog
import android.app.TimePickerDialog
import android.os.Bundle
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
import androidx.core.content.ContextCompat
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import androidx.core.view.WindowCompat
import com.example.mad_23012011026.DarkModeManager
import com.example.mad_23012011026.R
import java.text.SimpleDateFormat
import java.util.*

class AddReminderActivity : AppCompatActivity() {

    private lateinit var titleEditText: EditText
    private lateinit var dateEditText: EditText
    private lateinit var timeEditText: EditText
    private lateinit var categorySpinner: Spinner
    private lateinit var saveButton: Button
    private lateinit var darkModeButton: Button

    private val calendar = Calendar.getInstance()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Enable edge-to-edge
        SystemBarsHelper.enableEdgeToEdge(this)

        setContentView(R.layout.activity_add_reminder)

        setupWindowInsets()

        titleEditText = findViewById(R.id.titleEditText)
        dateEditText = findViewById(R.id.dateEditText)
        timeEditText = findViewById(R.id.timeEditText)
        categorySpinner = findViewById(R.id.categorySpinner)
        saveButton = findViewById(R.id.saveButton)
        darkModeButton = findViewById(R.id.darkModeButton)
```

Student Reminder App

```
setupCategorySpinner()
setupDatePicker()
setupTimePicker()
applyTheme()

saveButton.setOnClickListener {
    val title = titleEditText.text.toString().trim()
    val date = dateEditText.text.toString()
    val time = timeEditText.text.toString()
    val category = categorySpinner.selectedItem.toString()

    if (title.isEmpty() || date.isEmpty() || time.isEmpty()) {
        Toast.makeText(this, "Please fill all fields",
        Toast.LENGTH_SHORT).show()
    } else {
        val sharedPrefs = getSharedPreferences("reminders",
        MODE_PRIVATE)
        val reminderSet = sharedPrefs.getStringSet("reminderList",
        mutableSetOf())?.toMutableSet()

        val reminder = "$title\n$date at $time\nCategory: $category"
        reminderSet?.add(reminder)

        sharedPrefs.edit()
            .putStringSet("reminderList", reminderSet)
            .apply()

        Toast.makeText(this, "Reminder saved: $title",
        Toast.LENGTH_SHORT).show()
        finish()
    }
}

darkModeButton.setOnClickListener {
    DarkModeManager.toggleDarkMode(this)
    recreate()
}
}

private fun setupWindowInsets() {
    val mainLayout =
findViewById<androidx.constraintlayout.widget.ConstraintLayout>(R.id.mainLayout)
    SystemBarsHelper.setupMainContentInsets(mainLayout, 24)
}
```

Student Reminder App

```
private fun setupCategorySpinner() {
    val categories = arrayOf("Assignment", "Exam", "Class", "Personal")
    val adapter = ArrayAdapter(this,
        android.R.layout.simple_spinner_item, categories)

    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
    categorySpinner.adapter = adapter
}

private fun setupDatePicker() {
    val dateLayout = findViewById<LinearLayout>(R.id.dateLayout)
    dateLayout.setOnClickListener {
        val year = calendar.get(Calendar.YEAR)
        val month = calendar.get(Calendar.MONTH)
        val day = calendar.get(Calendar.DAY_OF_MONTH)

        val datePickerDialog = DatePickerDialog(this, { _, selectedYear, selectedMonth, selectedDay ->
            calendar.set(Calendar.YEAR, selectedYear)
            calendar.set(Calendar.MONTH, selectedMonth)
            calendar.set(Calendar.DAY_OF_MONTH, selectedDay)

            val sdf = SimpleDateFormat("dd/MM/yyyy", Locale.getDefault())
            dateEditText.setText(sdf.format(calendar.time))
        }, year, month, day)

        datePickerDialog.show()
    }
}

private fun setupTimePicker() {
    val timeLayout = findViewById<LinearLayout>(R.id.timeLayout)
    timeLayout.setOnClickListener {
        val hour = calendar.get(Calendar.HOUR_OF_DAY)
        val minute = calendar.get(Calendar.MINUTE)

        val timePickerDialog = TimePickerDialog(this, { _, selectedHour, selectedMinute ->
            calendar.set(Calendar.HOUR_OF_DAY, selectedHour)
            calendar.set(Calendar.MINUTE, selectedMinute)

            val sdf = SimpleDateFormat("HH:mm", Locale.getDefault())
            timeEditText.setText(sdf.format(calendar.time))
        }, hour, minute, true)

        timePickerDialog.show()
    }
}
```

Student Reminder App

```
        }

    }

private fun applyTheme() {
    val isDarkMode = DarkModeManager.isDarkMode(this)

    darkModeButton.text = if (isDarkMode) " Light Mode" else " Dark
Mode"

    val backgroundColor = if (isDarkMode)
        ContextCompat.getColor(this, R.color.darkBackgroundColor)
    else
        ContextCompat.getColor(this, R.color.white)

    findViewById<androidx.constraintlayout.widget.ConstraintLayout>(R.id.mainLayout).setBackgroundColor(backgroundColor)

    val textColor = if (isDarkMode)
        ContextCompat.getColor(this, R.color.darkEditTextTextColor)
    else
        ContextCompat.getColor(this, R.color.editTextColor)

    val hintColor = if (isDarkMode)
        ContextCompat.getColor(this, R.color.darkEditTextHintColor)
    else
        ContextCompat.getColor(this, R.color.editTextHintColor)

    titleEditText.setTextColor(textColor)
    titleEditText.setHintTextColor(hintColor)
    dateEditText.setTextColor(textColor)
    dateEditText.setHintTextColor(hintColor)
    timeEditText.setTextColor(textColor)
    timeEditText.setHintTextColor(hintColor)

    val editTextBackground = if (isDarkMode)
        ContextCompat.getDrawable(this, R.drawable.edit_text_bg_dark)
    else
        ContextCompat.getDrawable(this, R.drawable.edit_text_bg)

    titleEditText.background = editTextBackground
    dateEditText.background = editTextBackground
    timeEditText.background = editTextBackground
}
```

Student Reminder App

RegisterActivity.Kt

```
package com.example.mad_23012011026

import android.content.Intent
import android.os.Bundle
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
import androidx.core.content.ContextCompat

class RegisterActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        val passwordEditText =
            findViewById<EditText>(R.id.registerPasswordEditText)
        val confirmPasswordEditText =
            findViewById<EditText>(R.id.confirmPasswordEditText)
        val registerButton = findViewById<Button>(R.id.registerButton)
        val loginRedirectText =
            findViewById<TextView>(R.id.loginRedirectText)

        applyTheme()

        registerButton.setOnClickListener {
            val password = passwordEditText.text.toString()
            val confirmPassword = confirmPasswordEditText.text.toString()

            if (password != confirmPassword) {
                Toast.makeText(this, "Passwords do not match",
                    Toast.LENGTH_SHORT).show()
            } else {
                Toast.makeText(this, "Thank you for registration",
                    Toast.LENGTH_SHORT).show()

                val intent = Intent(this, MainActivity::class.java)
                startActivity(intent)
                finish()
            }
        }

        loginRedirectText.setOnClickListener {
            val intent = Intent(this, MainActivity::class.java)
        }
    }
}
```

Student Reminder App

```
        startActivity(intent)
        finish()
    }

}

private fun applyTheme() {
    val isDarkMode = DarkModeManager.isDarkMode(this)

    val backgroundColor = if (isDarkMode)
        ContextCompat.getColor(this, R.color.darkBackgroundColor)
    else
        ContextCompat.getColor(this, R.color.white)

    findViewById<androidx.constraintlayout.widget.ConstraintLayout>(R.id.mainLayout).setBackgroundColor(backgroundColor)

    val textColor = if (isDarkMode)
        ContextCompat.getColor(this, R.color.darkEditTextTextColor)
    else
        ContextCompat.getColor(this, R.color.editTextColor)

    val hintColor = if (isDarkMode)
        ContextCompat.getColor(this, R.color.darkEditTextHintColor)
    else
        ContextCompat.getColor(this, R.color.editTextHintColor)

    val passwordEditText =
findViewById<EditText>(R.id.registerPasswordEditText)
    val confirmPasswordEditText =
findViewById<EditText>(R.id.confirmPasswordEditText)

    passwordEditText.setTextColor(textColor)
    passwordEditText.setHintTextColor(hintColor)
    confirmPasswordEditText.setTextColor(textColor)
    confirmPasswordEditText.setHintTextColor(hintColor)

    val editTextBackground = if (isDarkMode)
        ContextCompat.getDrawable(this, R.drawable.edit_text_bg_dark)
    else
        ContextCompat.getDrawable(this, R.drawable.edit_text_bg)

    passwordEditText.background = editTextBackground
    confirmPasswordEditText.background = editTextBackground
}
```

Student Reminder App

SettingsActivity.Kt

```
package com.example.mad_23012011026

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.google.android.material.appbar.MaterialToolbar
import com.google.android.material/materialswitch.MaterialSwitch

class SettingsActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Enable edge-to-edge
        SystemBarsHelper.enableEdgeToEdge(this)

        setContentView(R.layout.activity_settings)

        setupWindowInsets()

        val toolbar =
            findViewById<MaterialToolbar>(R.id.settingsToolbar)
        setSupportActionBar(toolbar)
        supportActionBar?.setDisplayHomeAsUpEnabled(true)
        toolbar.setNavigationOnClickListener { finish() }

        val darkModeSwitch =
            findViewById<MaterialSwitch>(R.id.darkModeSwitch)
        darkModeSwitch.isChecked = DarkModeManager.isDarkMode(this)
        darkModeSwitch.setOnCheckedChangeListener { _, isChecked -
>
            DarkModeManager.setDarkMode(this, isChecked)
            recreate()
        }
    }

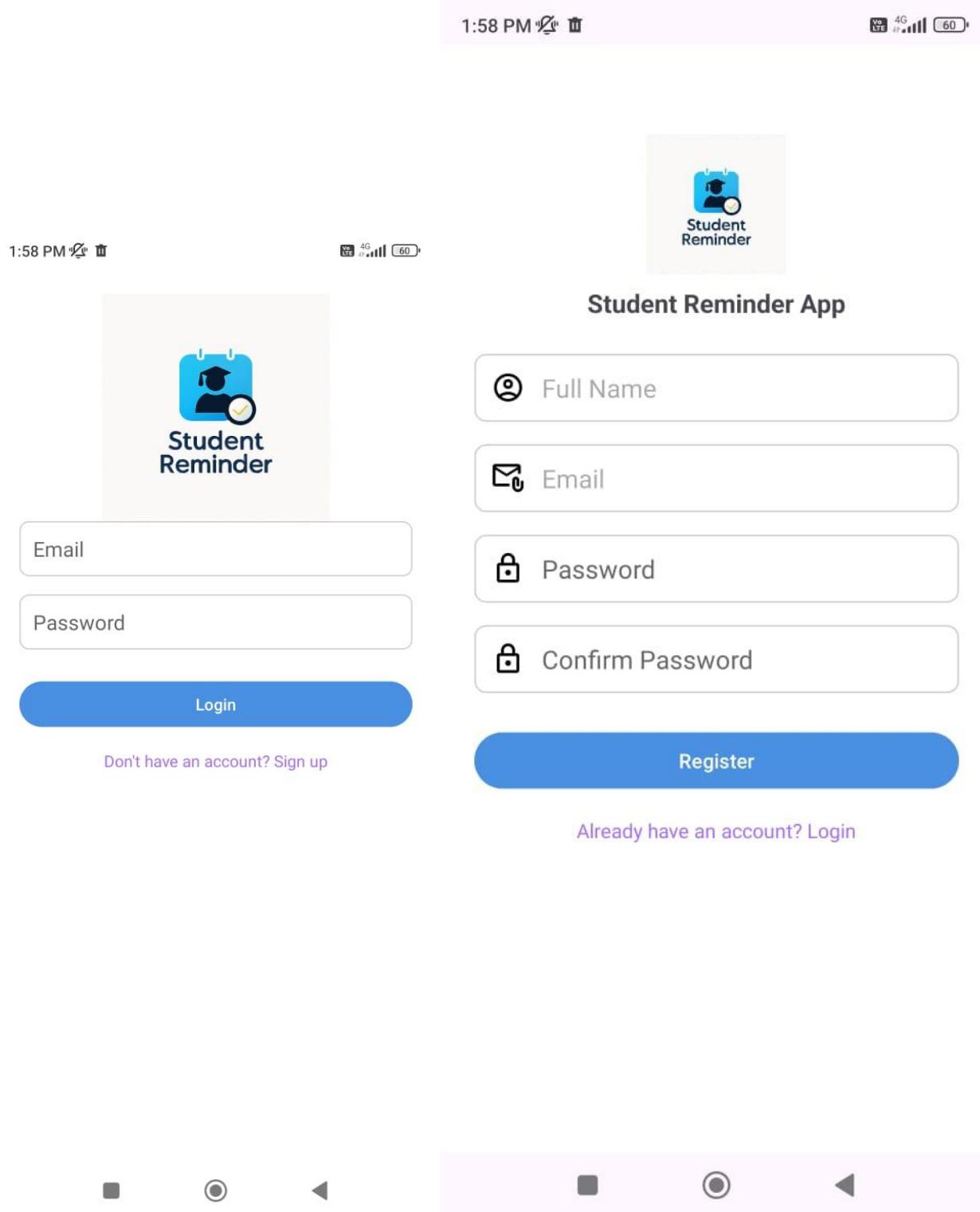
    private fun setupWindowInsets() {
        val toolbar =
            findViewById<MaterialToolbar>(R.id.settingsToolbar)
        val rootView =
```

Student Reminder App

```
findViewById<android.view.ViewGroup>(R.id.settingsMainLayout)

    SystemBarsHelper.setupSimpleActivityInsets(rootView,
toolbar)
}
}
```

Student Reminder App



Student Reminder App

