SURVEY APPLICATION

COMPOSE INPUT: A DEMONSTRATION OF TEXT INPUT AND VALIDATION WITH ANDROID

- ARUNPANDI R 715022104007

MANIMARAN V 715022104028

MOHANAPRASATH S 715022104031

RISHWAN A 715022104042

PROJECT OBJECTIVES:

- The goal of this project is to develop a user-friendly Android-based survey application that demonstrates text input and validation. The app is designed to provide a reliable platform for users to participate in surveys and for admins to manage survey data effectively. The objectives include:
- Enabling users to easily create accounts, log in, and participate in surveys.
- Ensuring smooth and secure access for administrators to view and manage survey results.
- Demonstrating form validation for both user and admin modules to enhance data integrity and user experience.

FUNCTIONALITIES:

This survey application will have two main modules: Admin Module and User Module, each with specific functionalities. The app's core functions include user registration, login with validation, and a survey-taking feature. Additionally, it includes hard-coded admin credentials to demonstrate an admin interface with restricted access.

OVERVIEW:

The survey app consists of multiple screens for both users and the admin, focusing on validation and secure login. Users can register, log in, and complete surveys, while the admin can log in with hardcoded credentials to view survey results.

DESCRIPTION:

- The application consists of the following interfaces:
- 1. Admin Module:
- Login Page: A login screen where the admin enters a hard-coded password ("admin") to gain access to the admin page. This page demonstrates text validation, allowing only specific credentials.
- Admin Page:
 - Accessible only after successful login with the correct credentials.
 - Displays the survey data and analytics collected from users.
 - Allows the admin to manage and view survey data (e.g., summary of responses).

2. User Module:

- Login Page: A login screen for registered users to enter their credentials and access the main survey interface. Includes validation for correct input format.
- Register Page: A registration screen for new users to create an account. It includes:
 - Username and password input with validation checks.
 - Error messages for invalid inputs (e.g., empty fields, password criteria).
- Main Page: The primary interface where users can access and take part in available surveys. This page shows the list of available surveys and records user responses.

Main Activity.kt:

package com.example.surveyapplication import android.content.Context import android.os.Bundle import androidx.activity.ComponentActivity import androidx.activity.compose.setContent import androidx.compose.foundation.Image import androidx.compose.foundation.layout.* import androidx.compose.material3.* 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.style.TextAlign
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
class MainActivity : ComponentActivity() {
  private lateinit var databaseHelper: SurveyDatabaseHelper
  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    databaseHelper = SurveyDatabaseHelper(this)
    setContent {
      FormScreen(this, databaseHelper)
```

```
@Composable
fun FormScreen(context: Context, databaseHelper: SurveyDatabaseHelper) {
Image(
    painterResource(id = R.drawable.background), contentDescription = "",
    alpha = 0.1F,
    contentScale = ContentScale.FillHeight,
    modifier = Modifier.padding(top = 40.dp)
// Define state for form fields
  var name by remember { mutableStateOf("") }
  var age by remember { mutableStateOf("") }
  var mobileNumber by remember { mutableStateOf("") }
  var genderOptions = listOf("Male", "Female", "Other")
  var selectedGender by remember { mutableStateOf("") }
  var error by remember { mutableStateOf("") }
```

```
Column(
   modifier = Modifier.padding(24.dp),
   horizontalAlignment = Alignment.Start,
   verticalArrangement = Arrangement.SpaceEvenly
 ) {
Text(
      fontSize = 36.sp,
      textAlign = TextAlign.Center,
      text = "Survey on Diabetics",
      color = Color(0xFF25b897)) Spacer(modifier = Modifier.height(24.dp))
Text(text = "Name :", fontSize = 20.sp)
   TextField(
      value = name,
      onValueChange = { name = it },
```

```
Spacer(modifier = Modifier.height(14.dp))
Text(text = "Age :", fontSize = 20.sp)
    TextField(
      value = age,
      onValueChange = { age = it },
Spacer(modifier = Modifier.height(14.dp))
Text(text = "Mobile Number :", fontSize = 20.sp)
    TextField(
      value = mobileNumber,
      onValueChange = { mobileNumber = it },
Spacer(modifier = Modifier.height(14.dp))
Text(text = "Gender :", fontSize = 20.sp)
    RadioGroup(
```

```
options = genderOptions,
      selectedOption = selectedGender,
      onSelectedChange = { selectedGender = it }
Spacer(modifier = Modifier.height(14.dp))
Text(text = "Diabetics :", fontSize = 20.sp)
RadioGroup(
     options = diabeticsOptions,
selectedOption = selectedDiabetics,
      onSelectedChange = { selectedDiabetics = it }
Text(
      text = error,
      textAlign = TextAlign.Center,
      modifier = Modifier.padding(bottom = 16.dp)
```

```
// Display Submit button
    Button(
      onClick = { if (name.isNotEmpty() && age.isNotEmpty() && mobileNumber.isNotEmpty() &&
genderOptions.isNotEmpty() && diabeticsOptions.isNotEmpty()) {
        val survey = Survey(
          id = null,
          name = name,
          age = age,
          mobileNumber = mobileNumber,
          gender = selectedGender,
          diabetics = selectedDiabetics
        databaseHelper.insertSurvey(survey)
        error = "Survey Completed"
```

```
} else {
         error = "Please fill all fields"
       colors = ButtonDefaults.buttonColors(containerColor = Color(0xFF84adb8)),
       modifier = Modifier.padding(start = 70.dp).size(height = 60.dp, width = 200.dp)
) {
       Text(text = "Submit")
@Composable
fun RadioGroup(
  options: List<String>,
  selectedOption: String?,
```

```
onSelectedChange: (String) -> Unit
) {
  Column {
    options.forEach { option ->
      Row(
         Modifier
           .fillMaxWidth()
           .padding(horizontal = 5.dp)
      ) {
         RadioButton(
           selected = option == selectedOption,
           onClick = { onSelectedChange(option) }
         Text(
           text = option,
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"</pre>
  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.SurveyApplication"
    tools:targetApi="31">
    <activity
```

```
android:name=".RegisterActivity"
     android:exported="false"
     android:label="@string/title_activity_register"
     android:theme="@style/Theme.SurveyApplication" />
   <activity
     android:name=".MainActivity"
     android:exported="false"
     android:label="MainActivity"
     android:theme="@style/Theme.SurveyApplication" />
   <activity
     android:name=".AdminActivity"android:exported="false"
     android:label="@string/title activity admin"
     android:theme="@style/Theme.SurveyApplication" />
   <activity
     android:name=".LoginActivity"
```

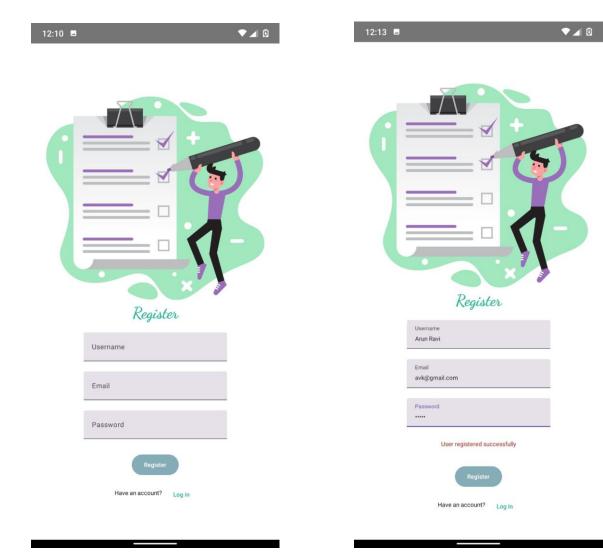
```
android:exported="true"
      android:label="@string/app_name"
      android:theme="@style/Theme.SurveyApplication">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

USER MODULE:

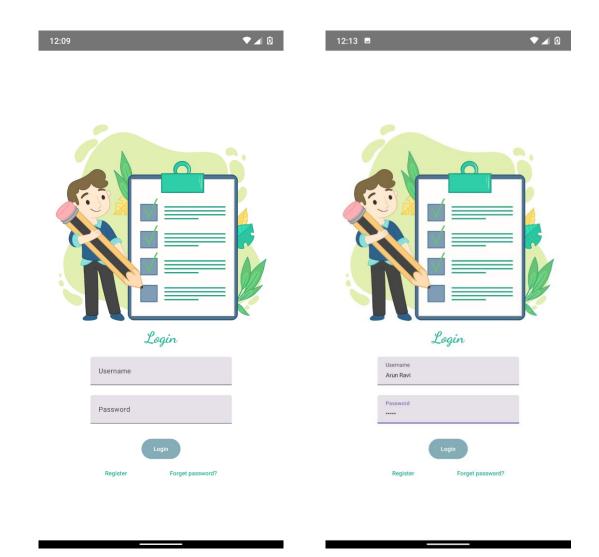
USER INTERFACE:



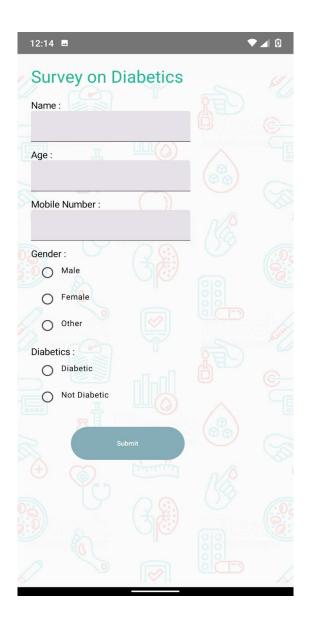
REGISTER PAGE:

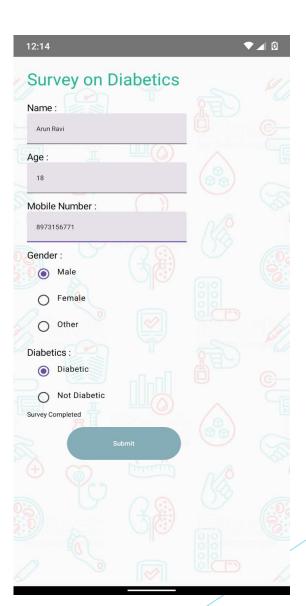


LOGIN PAGE:



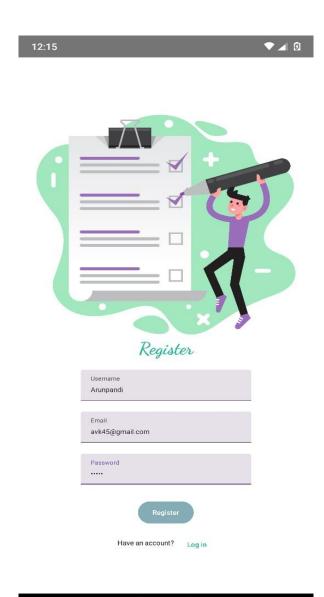
MAIN PAGE FOR USER MODULE:





ADMIN MODULE:

REGISTER PAGE:



LOGIN PAGE:



MAIN PAGE FOR ADMIN MODULE:

