Aim: To include icons, images, fonts in Flutter app.

Theory:

A form in Flutter is a collection of input fields that allow users to enter and submit data. The Form widget provides a way to validate user input before processing it. The TextFormField widget is commonly used within a form because it supports validation, unlike TextField. A GlobalKey is assigned to the form to track its state and check if all inputs are valid before submission.

Implementation in Our Code

In our Woman Safety App, we created a login form using the Form widget. The form includes:

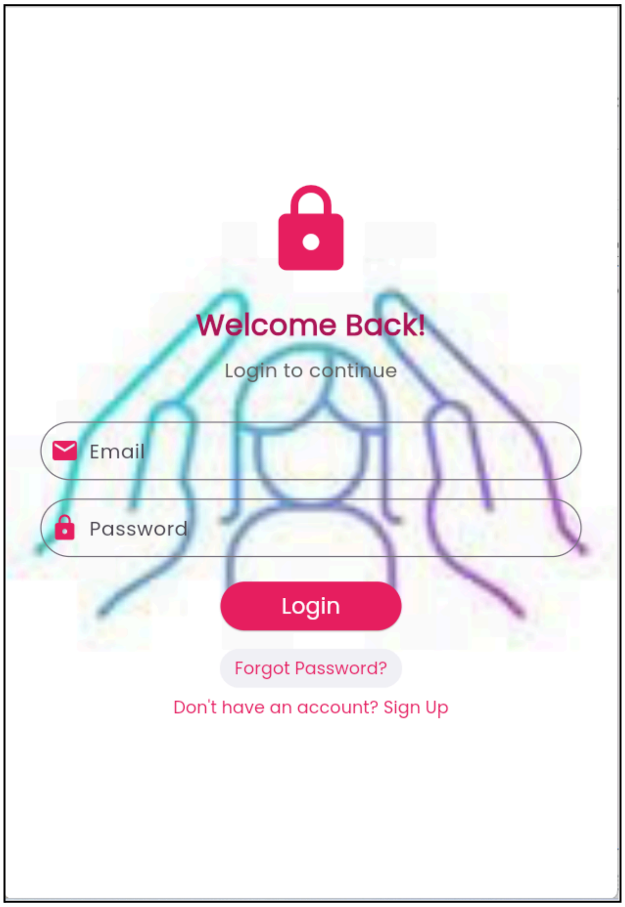
- Email Input Field – Accepts an email and ensures it follows the correct format.
- Password Input Field – Accepts a password and checks that it has at least six characters.
- Validation – If any field is empty or incorrect, an error message is displayed.
- Login Button – Validates input before navigating to the HomeScreen.

Code :

login.dart

```
import 'package:flutter/material.dart';
import 'main.dart'; // Import main.dart to navigate to HomeScreen

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final _formKey = GlobalKey<FormState>(); // Key to track form state
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();

  void _login() {
    if (_formKey.currentState!.validate()) {
      // If the form is valid, navigate to HomeScreen
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => HomeScreen()),
      );
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: [
          Positioned.fill(
            child: Image.asset(
              'assets/bg.png', // Background image
              fit: BoxFit.cover,
            ),
          ),
          Center(
            child: Padding(
              padding: EdgeInsets.symmetric(horizontal: 30),
              child: Form(
                key: _formKey, // Assign form key
                child: Column(
                  mainAxisSize: MainAxisSize.min,
                  children: [
                    Icon(Icons.lock, size: 80, color: Colors.pink),
                    SizedBox(height: 20),
                    Text(
                      "Welcome Back!",
                      style: TextStyle(
                        fontSize: 24,
                        fontWeight: FontWeight.bold,
                        color: Colors.pink.shade800,
                      ),
                    ),
                    SizedBox(height: 10),
                    Text(
                      "Login to continue",
                      style: TextStyle(fontSize: 16, color: Colors.grey[700]),
                    ),
                    SizedBox(height: 30),

                    // Email Input
                    TextFormField(
                      controller: _emailController,
                      decoration: InputDecoration(
                        labelText: "Email",
                        prefixIcon: Icon(Icons.email, color: Colors.pink),
                        border: OutlineInputBorder(
                          borderRadius: BorderRadius.circular(30),
                        ),
                      ),
                      validator: (value) {
                        if (value == null || value.isEmpty) {
                          return "Please enter your email";
                        } else if
(!RegExp(r"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$")
                            .hasMatch(value)) {
                          return "Enter a valid email";
                        }
                        return null;
                      },
                    ),
                    SizedBox(height: 15),

                    // Password Input
                    TextFormField(
                      controller: _passwordController,
                      obscureText: true,
                      decoration: InputDecoration(
                        labelText: "Password",
                        prefixIcon: Icon(Icons.lock, color: Colors.pink),
                        border: OutlineInputBorder(
                          borderRadius: BorderRadius.circular(30),
                        ),
                      ),
                      validator: (value) {
                        if (value == null || value.isEmpty) {
                          return "Please enter your password";
                        } else if (value.length < 6) {
                          return "Password must be at least 6 characters";
                        }
                        return null;
                      },
                    ),
                    SizedBox(height: 20),

                    // Login Button
                    ElevatedButton(
                      onPressed: _login,
                      style: ElevatedButton.styleFrom(
                        backgroundColor: Colors.pink,
                        padding: EdgeInsets.symmetric(horizontal: 50, vertical:
15),

                        shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(30),
                        ),
                      ),
                      child: Text(
                        "Login",
                        style: TextStyle(fontSize: 18, color: Colors.white),
                      ),
                    ),

                    // Sign Up & Forgot Password
                    SizedBox(height: 15),
                    TextButton(
                      onPressed: () {
                        print("Forgot Password?");
                      },
                      child: Text("Forgot Password?", style: TextStyle(color:
Colors.pink)),
                    ),
                    TextButton(
                      onPressed: () {
                        print("Go to Sign Up");
                      },
                      child: Text("Don't have an account? Sign Up", style:
TextStyle(color: Colors.pink)),
                    ),
                  ],
                ),
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```

Screenshot:



Conclusion: In this experiment, we successfully implemented an interactive login form using the Form widget with validation for email and password. Initially, we faced issues with validating user input and preventing navigation on invalid credentials, but we resolved them by using TextFormField, adding validation logic, and checking the form state before allowing login.