NAME: Bhushan Malpani DIV: D15C ROLL NO: 33

Experiment 03 : To include icons, images, fonts in Flutter app

Theory:

A visually appealing and user-friendly interface is key to enhancing the user experience (UX) in Flutter applications. In your Quiz App's SignUpPage, you've effectively used icons, custom images, and fonts to create an engaging onboarding experience. Here's how each plays a role:

1. Icons

- Purpose: Icons provide visual cues, improve navigation, and enhance form usability.
- Used in input fields for username, email, password, and role to indicate the type of input required.
- Widget Used: Icon, Icons.person, Icons.email, Icons.lock, etc

2. Images

- Purpose: Images (like logos or banners) reinforce branding and break monotony in the UI.
- Displays the Quizzify logo at the top of the sign-up screen.

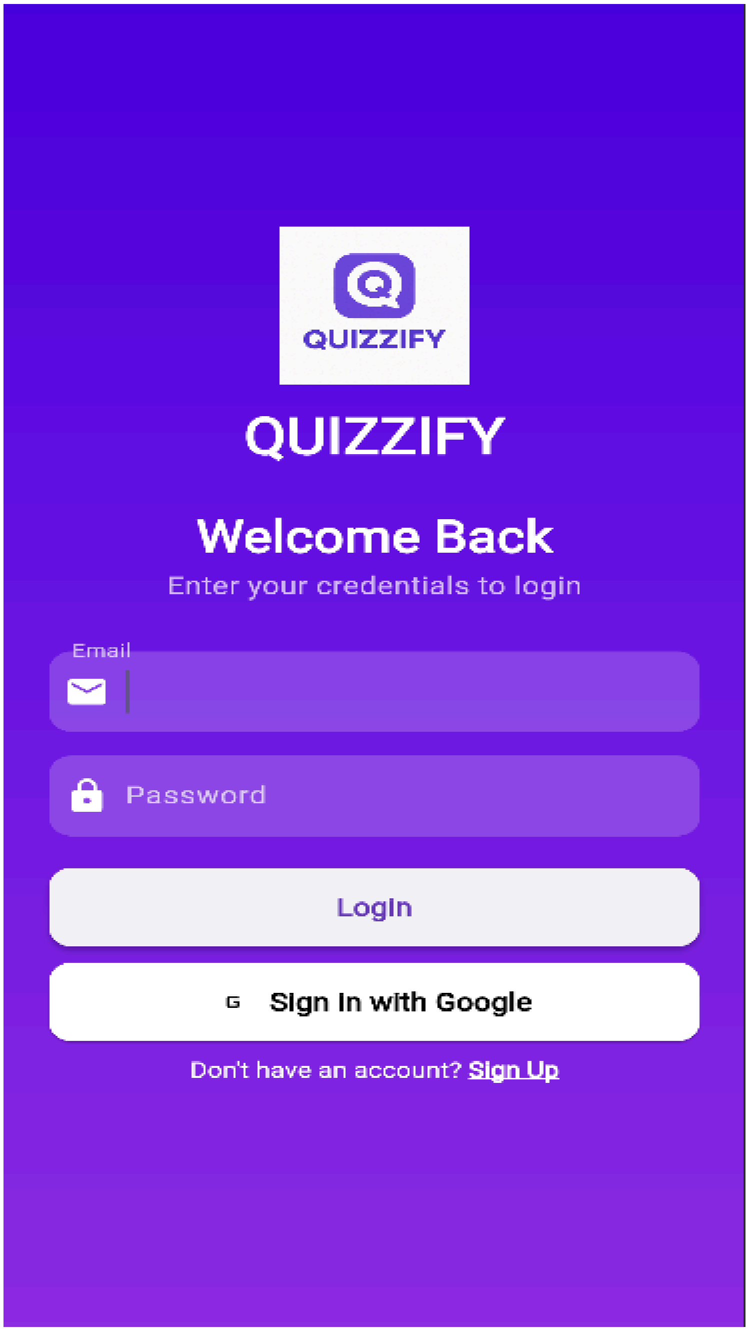
3. Fonts

- Purpose: Custom fonts give your app a unique visual identity and improve readability.
- Aligns typography with the app's theme.
- Helps differentiate important texts (like headings and labels).

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/gestures.dart';
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'homepage.dart';
import 'signuppage.dart';
import 'admin.dart';
import 'forgot_page.dart';

class SignInPage extends StatefulWidget {
  const SignInPage({super.key});

  @override
  State<SignInPage> createState() => _SignInPageState();
}
```

```
class _SignInPageState extends State<SignInPage> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final _formKey = GlobalKey<FormState>();
  bool _isLoading = false;

  Future<void> _signIn() async {
    if (!_formKey.currentState!.validate()) {
      return; // Don't proceed if form validation fails
    }

    setState(() {
      _isLoading = true;
    });

    try {
      UserCredential userCredential = await _auth.signInWithEmailAndPassword(
        email: _emailController.text.trim(),
        password: _passwordController.text.trim(),
      );

      DocumentSnapshot userDoc = await FirebaseFirestore.instance
          .collection('users')
          .doc(userCredential.user!.uid)
          .get();

      if (userDoc.exists) {
        String role = userDoc['role'];

        if (role == 'admin') {
          Navigator.pushReplacement(
            context,
            MaterialPageRoute(builder: (context) => const AdminPage()),
          );
        } else {
          Navigator.pushReplacement(
            context,
            MaterialPageRoute(builder: (context) => const HomePage()),
          );
        }
      }
    } on FirebaseAuthException catch (e) {
      _showErrorDialog(e);
    } catch (e) {
      _showErrorDialog(e);
    }

    setState(() {
      _isLoading = false;
    });
  }

  // Show Alert Dialog for Errors
  void _showErrorDialog(dynamic error) {
    String errorMessage = "An error occurred. Please try again.";

    if (error is FirebaseAuthException) {
      switch (error.code) {
        case "invalid-email":
          errorMessage = "Invalid email format.";
          break;
        case "user-not-found":
          errorMessage = "No account found with this email.";
          break;
        case "wrong-password":
          errorMessage = "Incorrect password. Please try again.";
          break;
        case "user-disabled":
          errorMessage = "This account has been disabled.";
          break;
        case "too-many-requests":
          errorMessage = "Too many failed attempts. Please try again later.";
          break;
        default:
          errorMessage = "Login failed: ${error.message}";
      }
    } else {
      errorMessage = "Unexpected error: ${error.toString()}";
    }

    showDialog(
      context: context,
      builder: (context) => AlertDialog(
        title: const Text("Login Error"),
        content: Text(errorMessage),
        actions: [
          TextButton(
            onPressed: () => Navigator.pop(context),
            child: const Text("OK"),
          ),
        ],
      ),
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        decoration: const BoxDecoration(
          gradient: LinearGradient(
            colors: [Color(0xFF4A00E0), Color(0xFF8E2DE2)],
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
          ),
        ),
        child: Center(
          child: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 24),
            child: Form(
              key: _formKey,
              child: Column(
                mainAxisSize: MainAxisSize.min,
                children: [
                  // App Icon
                  Image.asset(
                    "assets/images/quizzify_logo.png", // Ensure this image is in the assets
folder
```

```
  height: 100,
),
const SizedBox(height: 10),

// App Name
const Text(
  "QUIZZIFY",
  style: TextStyle(
    fontSize: 32,
    fontWeight: FontWeight.bold,
    color: Colors.white,
  ),
),
const SizedBox(height: 20),

const Text(
  "Welcome Back",
  style: TextStyle(
    fontSize: 28,
    fontWeight: FontWeight.bold,
    color: Colors.white,
  ),
),
const SizedBox(height: 5),
const Text(
  "Enter your credentials to login",
  style: TextStyle(fontSize: 16, color: Colors.white70),
),
const SizedBox(height: 30),

// Email Field
TextFormField(
  controller: _emailController,
  keyboardType: TextInputType.emailAddress,
  decoration: InputDecoration(
    labelText: "Email",
    labelStyle:
        const TextStyle(color: Colors.white70, fontSize: 16),
    filled: true,
    fillColor: Colors.white.withOpacity(0.2),
    prefixIcon: const Icon(Icons.email, color: Colors.white),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: BorderSide.none,
    ),
  ),
  style: const TextStyle(color: Colors.white, fontSize: 18),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return "Please enter your email";
    } else if (!value.contains('@') || !value.contains('.')) {
      return "Enter a valid email";
    }
    return null;
  },
),

const SizedBox(height: 15),

// Password Field
TextFormField(
  controller: _passwordController,
  obscureText: true,
  decoration: InputDecoration(
    labelText: "Password",
    labelStyle:
        const TextStyle(color: Colors.white70, fontSize: 16),
    filled: true,
    fillColor: Colors.white.withOpacity(0.2),
    prefixIcon: const Icon(Icons.lock, color: Colors.white),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: BorderSide.none,
    ),
  ),
  style: const TextStyle(color: Colors.white, fontSize: 18),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return "Please enter your password";
    } else if (value.length < 4 || value.length > 8) {
      return "Password must be 4-8 characters";
    }
    return null;
  },
),

const SizedBox(height: 20),

// Login Button
SizedBox(
  width: double.infinity,
  height: 50,
  child: ElevatedButton(
    onPressed: _isLoading ? null : _signIn,
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.white,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(12),
      ),
    ),
    child: _isLoading
        ? const CircularProgressIndicator(
            valueColor: AlwaysStoppedAnimation<Color>(
                Colors.deepPurple),
          )
        : const Text(
            "Login",
            style: TextStyle(
              color: Colors.deepPurple,
              fontSize: 16,
              fontWeight: FontWeight.bold,
            ),
          ),
  ),
),
const SizedBox(height: 15),

// Forgot Password
Align(
  alignment: Alignment.centerRight,
  child: TextButton(
    onPressed: () {
      Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) => ForgotPasswordPage()),
      );
    },
    child: const Text(
      "Forgot Password?",
      style: TextStyle(
        color: Colors.white,
        decoration: TextDecoration.underline,
      ),
    ),
  ),
),

// Sign-Up Navigation
RichText(
  text: TextSpan(
    style: const TextStyle(fontSize: 14, color: Colors.white),
    children: [
      const TextSpan(text: "Don't have an account? "),
      TextSpan(
        text: "Sign Up",
        style: const TextStyle(
          color: Colors.white,
          fontWeight: FontWeight.bold,
          decoration: TextDecoration.underline,
        ),
        recognizer: TapGestureRecognizer()
          ..onTap = () {
            Navigator.push(
              context,
              MaterialPageRoute(
                  builder: (context) => const SignUpPage()),
            );
          },
      ),
                      ],
                    ),
                  ),
                  const SizedBox(height: 20),
                ],
              ),
            ),
          ),
        ),
      ),
    );
  }
}
```



QUIZZIFY

Welcome Back

Enter your credentials to login

Don't have an account? Sign Up

Conclusion:

In the SignUpPage of your Quiz App, you've effectively demonstrated the use of icons, images, and fonts—three key UI elements that elevate the look and feel of a Flutter app.

Icons provide intuitive form navigation.
Images add branding and engagement.
Fonts (default or custom) improve visual hierarchy and tone.