Main.dart

```dart
import 'package:auth/screens/SplashScreen.dart';
import 'package:flutter/material.dart';
import 'screens/HomeScreen.dart';
import 'screens/CategoryScreen.dart';
import 'screens/ProductScreen.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();

  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Project',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor:
Colors.orangeAccent),
        useMaterial3: true,
      ),
      home: const SplashScreen(),
      routes: {
        '/home': (context) => const HomeScreen(),
        '/categories': (context) => const CategoryScreen(),
        '/products': (context) => const ProductScreen(),

      },
    );
  }
}
```

Constants/constants.dart

```dart
class ApiConstants {
  static const String baseUrl = 'http://192.168.0.229/sjec';
}
```

screens/SplashScreen.dart

```dart
import 'package:flutter/material.dart';
import 'package:auth/screens/LoginScreen.dart';
```

```dart
import 'package:flutter/services.dart';
import 'package:animated_text_kit/animated_text_kit.dart';

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

  @override
  State<SplashScreen> createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> with
SingleTickerProviderStateMixin {
  late AnimationController _controller;
  late Animation<double> _fadeInAnimation;
  late Animation<double> _scaleAnimation;

  @override
  void initState() {
    super.initState();

    // Set up full-screen immersive mode for splash screen

SystemChrome.setEnabledSystemUIMode(SystemUiMode.immersive);

    // Initialize animations
    _controller = AnimationController(
      duration: const Duration(milliseconds: 2000),
      vsync: this,
    );

    _fadeInAnimation = Tween<double>(begin: 0.0, end:
1.0).animate(
      CurvedAnimation(
        parent: _controller,
        curve: const Interval(0.0, 0.5, curve: Curves.easeIn),
      ),
    );

    _scaleAnimation = Tween<double>(begin: 0.8, end:
1.0).animate(
      CurvedAnimation(
        parent: _controller,
        curve: const Interval(0.0, 0.5, curve:
Curves.easeOutCubic),
      ),
    );

    _controller.forward();

    // Navigate to login screen after animation completes
    Future.delayed(const Duration(milliseconds: 3500), () {
```

```
      // Restore system UI before navigating
SystemChrome.setEnabledSystemUIMode(SystemUiMode.edgeToEdge);
      Navigator.pushReplacement(
        context,
        PageRouteBuilder(
          pageBuilder: (context, animation,
secondaryAnimation) => const LoginScreen(),
          transitionsBuilder: (context, animation,
secondaryAnimation, child) {
            var fadeTransition = Tween<double>(begin: 0.0,
end: 1.0).animate(animation);
            return FadeTransition(opacity: fadeTransition,
child: child);
          },
          transitionDuration: const Duration(milliseconds:
800),
        ),
      );
    });
  }

  @override
  void dispose() {
    _controller.dispose();
SystemChrome.setEnabledSystemUIMode(SystemUiMode.edgeToEdge);
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        decoration: BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
            colors: [
              Colors.blue.shade900,
              Colors.blue.shade700,
              Colors.blue.shade500,
            ],
          ),
        ),
        child: AnimatedBuilder(
          animation: _controller,
          builder: (context, child) {
            return FadeTransition(
              opacity: _fadeInAnimation,
```

```
              child: ScaleTransition(
                scale: _scaleAnimation,
                child: Center(
                  child: Column(
                    mainAxisAlignment:
MainAxisAlignment.center,
                    children: [
                      // Logo with shadow
                      Container(
                        decoration: BoxDecoration(
                          shape: BoxShape.circle,
                          boxShadow: [
                            BoxShadow(
                              color:
Colors.black.withOpacity(0.3),
                              blurRadius: 15,
                              offset: const Offset(0, 8),
                            ),
                          ],
                        ),
                        child: ClipRRect(
                          borderRadius:
BorderRadius.circular(75),
                          child: Image.asset(
                            'assets/splash_image.jpg',
                            width: 150,
                            height: 150,
                            fit: BoxFit.cover,
                          ),
                        ),
                      ),
                      const SizedBox(height: 30),

                      // Animated app name
                      AnimatedTextKit(
                        animatedTexts: [
                          FadeAnimatedText(
                            'Project Name',
                            textStyle: const TextStyle(
                              fontSize: 32,
                              fontWeight: FontWeight.bold,
                              color: Colors.white,
                              letterSpacing: 1.5,
                            ),
                            duration: const
Duration(milliseconds: 2000),
                          ),
                        ],
                        totalRepeatCount: 1,
                      ),
```

```
                        const SizedBox(height: 15),

                        // Subtitle
                        const Text(
                          'Make a Difference Today',
                          style: TextStyle(
                            fontSize: 16,
                            color: Colors.white70,
                            fontWeight: FontWeight.w500,
                          ),
                        ),

                        const SizedBox(height: 50),

                        // Loading indicator
                        SizedBox(
                          width: 40,
                          height: 40,
                          child: CircularProgressIndicator(
                            valueColor:
AlwaysStoppedAnimation<Color>(Colors.white.withOpacity(0.8)),
                            strokeWidth: 3,
                          ),
                        ),
                      ],
                    ),
                  ),
                ),
              );
            },
          ),
        ),
      );
  }
}
```

RegisterScreen.dart

```
import 'package:auth/screens/LoginScreen.dart';
import 'package:flutter/material.dart';

import '../service/RegisterService.dart';

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

  @override
  State<RegisterScreen> createState() =>
_RegisterScreenState();
}
```

```dart
class _RegisterScreenState extends State<RegisterScreen> {
  final _formKey = GlobalKey<FormState>();
  final _nameController = TextEditingController();
  final _emailController = TextEditingController();
  final _numberController = TextEditingController();
  final _passwordController = TextEditingController();

  bool _isLoading = false;

  Future<void> _register() async {
    if (_formKey.currentState!.validate()) {
      setState(() {
        _isLoading = true;
      });

      try {
        final data = await AuthService.register(
          name: _nameController.text,
          email: _emailController.text,
          number: _numberController.text,
          password: _passwordController.text,
        );

        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text(data['message'])),
        );

        if (data['success']) {
          Navigator.pop(context);
        }
      } catch (e) {
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(content: Text('An error occurred')),
        );
      } finally {
        setState(() {
          _isLoading = false;
        });
      }
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Register', style: TextStyle(color:
Colors.white)),
        flexibleSpace: Container(
          decoration: BoxDecoration(
```

```
            gradient: LinearGradient(
              colors: [Colors.blue.shade900,
Colors.blue.shade700],
              begin: Alignment.topCenter,
              end: Alignment.bottomCenter,
            ),
          ),
        ),
      ),
      body: SingleChildScrollView(
        padding: const EdgeInsets.all(16.0),
        child: Form(
          key: _formKey,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
              Card(
                elevation: 5,
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(16),
                ),
                child: Padding(
                  padding: const EdgeInsets.all(16.0),
                  child: Column(
                    children: [
                       Image.asset(

'assets/register_image.jpg',
                                                  height: 250,
                                                  width: 450,
                                              ),
                      TextFormField(
                        controller: _nameController,
                        decoration: const InputDecoration(
                          labelText: 'Name',
                          border: OutlineInputBorder(),
                        ),
                        validator: (value) => value == null ||
value.isEmpty
                            ? 'Please enter your name'
                            : null,
                      ),
                      const SizedBox(height: 16),
                      TextFormField(
                        controller: _emailController,
                        decoration: const InputDecoration(
                          labelText: 'Email',
                          border: OutlineInputBorder(),
                        ),
                        validator: (value) => value == null ||
value.isEmpty
```

```
                            ? 'Please enter your email'
                            : null,
                      ),
                      const SizedBox(height: 16),
                      TextFormField(
                        controller: _numberController,
                        decoration: const InputDecoration(
                          labelText: 'Phone Number',
                          border: OutlineInputBorder(),
                        ),
                        validator: (value) => value == null ||
value.isEmpty
                            ? 'Please enter your phone number'
                            : null,
                      ),
                      const SizedBox(height: 16),
                      TextFormField(
                        controller: _passwordController,
                        decoration: const InputDecoration(
                          labelText: 'Password',
                          border: OutlineInputBorder(),
                        ),
                        obscureText: true,
                        validator: (value) => value == null ||
value.isEmpty
                            ? 'Please enter your password'
                            : null,
                      ),
                    ],
                  ),
                ),
              ),
              const SizedBox(height: 24),
              _isLoading
                  ? const Center(child:
CircularProgressIndicator())
                  : ElevatedButton(
                      onPressed: _register,
                      style: ElevatedButton.styleFrom(
                        backgroundColor: Colors.blue.shade700,
                        shape: RoundedRectangleBorder(
                          borderRadius:
BorderRadius.circular(8),
                        ),
                      ),
                      child: const Text('Register', style:
TextStyle(color: Colors.white)),
                    ),
            ],
          ),
        ),
```

```
      ),
    );
  }
}
```

loginScreen.dart

```
import 'package:auth/screens/ForgotPasswordScreen.dart';
import 'package:flutter/material.dart';
import 'package:auth/screens/HomeScreen.dart';
import 'package:auth/screens/RegisterScreen.dart';
import 'package:shared_preferences/shared_preferences.dart';
import '../service/LoginService.dart';

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

    @override
    State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
    final _formKey = GlobalKey<FormState>();
    final _emailController = TextEditingController();
    final _passwordController = TextEditingController();

    @override
    void initState() {
        super.initState();
        _checkLoginStatus();
    }

    Future<void> _checkLoginStatus() async {
        final prefs = await SharedPreferences.getInstance();
        final isLoggedIn = prefs.getBool('isLoggedIn') ??
false;

        if (isLoggedIn) {
            Navigator.pushReplacement(
                context,
                MaterialPageRoute(builder: (context) => const
HomeScreen()),
            );
        }
    }

    Future<void> _login() async {
        if (_formKey.currentState!.validate()) {
            try {
                final data = await AuthService.login(
```

```
                    _emailController.text,
                    _passwordController.text,
                );

                if (data['success']) {
                    final prefs = await
SharedPreferences.getInstance();
                    await prefs.setBool('isLoggedIn', true);
                    await prefs.setString('userId',
data['id'].toString());
                    await prefs.setString('name',
data['name']);
                    await prefs.setString('email',
data['email']);

                    Navigator.pushReplacement(
                        context,
                        MaterialPageRoute(builder: (context)
=> const HomeScreen()),
                    );
                } else {

ScaffoldMessenger.of(context).showSnackBar(
                        SnackBar(content:
Text(data['message'])),
                    );
                }
            } catch (e) {
                ScaffoldMessenger.of(context).showSnackBar(
                    SnackBar(content: Text('An error occurred:
${e.toString()}')),
                );
            }
        }
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: const Text('Login', style:
TextStyle(color: Colors.white)),
                flexibleSpace: Container(
                    decoration: BoxDecoration(
                        gradient: LinearGradient(
                            colors: [Colors.blue.shade900,
Colors.blue.shade700],
                            begin: Alignment.topCenter,
                            end: Alignment.bottomCenter,
                        ),
                    ),
```

```
                ),
            ),
            body:Align(
        alignment: Alignment.center,
        child:
            SingleChildScrollView(
                padding: const EdgeInsets.all(16.0),
                child:Align(
                    alignment: Alignment.center,
                    child:
                Form(
                    key: _formKey,
                    child: Column(
                        crossAxisAlignment:
CrossAxisAlignment.stretch,
                        mainAxisAlignment:
MainAxisAlignment.center,
                        children: [
                            Card(
                                elevation: 5,
                                shape: RoundedRectangleBorder(
                                    borderRadius:
BorderRadius.circular(16),
                                ),
                                child: Padding(
                                    padding: const
EdgeInsets.all(16.0),
                                    child: Column(
                                        children: [
                                            Image.asset(

'assets/login_image.jpg',
                                                height: 250,
                                                width: 450,
                                            ),
                                            const
SizedBox(height: 16),
                                            TextFormField(
                                                controller:
_emailController,
                                                decoration:
const InputDecoration(
                                                    labelText:
'Email',
                                                    border:
OutlineInputBorder(),
                                                ),
                                                validator:
(value) =>
                                                    value
== null || value.isEmpty ? 'Please enter your email' : null,
```

```
                                                ),
                                                const
SizedBox(height: 16),
                                                TextFormField(
                                                    controller:
_passwordController,
                                                    decoration:
const InputDecoration(
                                                        labelText:
'Password',
                                                        border:
OutlineInputBorder(),
                                                    ),
                                                    obscureText:
true,
                                                    validator:
(value) =>
                                                            value
== null || value.isEmpty ? 'Please enter your password' :
null,
                                                ),
                                                const
SizedBox(height: 24),
                                                ElevatedButton(
                                                    onPressed:
_login,
                                                    style:
ElevatedButton.styleFrom(

backgroundColor: Colors.blue.shade700,
                                                        shape:
RoundedRectangleBorder(

borderRadius: BorderRadius.circular(8),
                                                        ),
                                                    ),
                                                    child: const
Text('Login', style: TextStyle(color: Colors.white)),
                                                ),
                                                TextButton(
                                                    onPressed: ()
{

Navigator.push(

context,

MaterialPageRoute(builder: (context) => const
RegisterScreen()),
                                                        );
                                                    },
```

```
                                                    child: const
Text('Don\'t have an account? Register'),
                                                ),
                                                TextButton(
                                                    onPressed: ()
{

Navigator.push(

context,

MaterialPageRoute(builder: (context) => const
ForgotPasswordScreen()),
                                                        );
                                                },
                                                    child: const
Text('Forgot Password?'),
                                                ),
                                            ],
                                        ),
                                    ),
                                ),
                            ],
                        ),
                        ),
                    ),
                ),
            ),
        );
    }
}
```

HomeScreen.dart

```
import 'package:carousel_slider/carousel_slider.dart';

import 'package:auth/screens/ChangePassword.dart';

import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:auth/screens/LoginScreen.dart';
import 'package:flutter/services.dart';
import '../widgets/AppDrawer.dart';

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

    @override
    State<HomeScreen> createState() => _HomeScreenState();
}
```

```dart
class _HomeScreenState extends State<HomeScreen> {
    int _currentCarouselIndex = 0;
    final GlobalKey<ScaffoldState> _scaffoldKey =
GlobalKey<ScaffoldState>();

    Future<void> _logout(BuildContext context) async {
        final confirmed = await showDialog<bool>(
            context: context,
            builder: (BuildContext context) {
                return AlertDialog(
                    shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(16)),
                    title: const Text('Logout'),
                    content: const Text('Are you sure you want
to logout?'),
                    actions: <Widget>[
                        TextButton(
                            child: const Text('Cancel'),
                            onPressed: () =>
Navigator.of(context).pop(false),
                        ),
                        ElevatedButton(
                            style: ElevatedButton.styleFrom(
                                backgroundColor:
Theme.of(context).primaryColor,
                                foregroundColor: Colors.white,
                                shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(8)),
                            ),
                            child: const Text('Logout'),
                            onPressed: () =>
Navigator.of(context).pop(true),
                        ),
                    ],
                );
            },
        );

        if (confirmed == true) {
            final prefs = await
SharedPreferences.getInstance();
            await prefs.setBool('isLoggedIn', false);

            if (!mounted) return;

            Navigator.pushReplacement(
                context,
                MaterialPageRoute(builder: (context) => const
```

```
LoginScreen()),
            );
        }
    }

    @override
    Widget build(BuildContext context) {
        // Set status bar color to match app theme
        SystemChrome.setSystemUIOverlayStyle(
            SystemUiOverlayStyle(
                statusBarColor: Colors.transparent,
                statusBarIconBrightness: Brightness.light,
            ),
        );

        final List<Map<String, dynamic>> gridItems = [
            {
                'title': 'feature-1',
                'icon': Icons.pets,
                'color': Colors.blue.shade700,
            },
            {
                'title': 'feature-2',
                'icon': Icons.favorite,
                'color': Colors.red.shade700,
            },
            {
                'title': 'feature-3',
                'icon': Icons.add_circle,
                'color': Colors.green.shade700,
            },
            {
                'title': 'feature-4',
                'icon': Icons.volunteer_activism,
                'color': Colors.orange.shade700,
            },
        ];

        final List<Map<String, dynamic>> carouselItems = [
            {
                'image': 'assets/image1.jpg',
                'title': 'Help Dogs Find Homes',
            },
            {
                'image': 'assets/image2.jpg',
                'title': 'Support Our Rescues',
            },
            {
                'image': 'assets/image3.jpg',
                'title': 'Make A Difference Today',
            },
```

```
        ];

        return Scaffold(
            key: _scaffoldKey,
            extendBodyBehindAppBar: true,
            appBar: AppBar(
                elevation: 0,
                backgroundColor: Colors.transparent,
                foregroundColor: Colors.white,
                title: const Text(
                    'Project Name',
                    style: TextStyle(
                        fontSize: 22,
                        fontWeight: FontWeight.bold,
                    ),
                ),
                leading: IconButton(
                    icon: const Icon(Icons.menu),
                    onPressed: () =>
_scaffoldKey.currentState?.openDrawer(),
                ),
                actions: [
                    IconButton(
                        icon: const Icon(Icons.logout),
                        onPressed: () => _logout(context),
                        tooltip: 'Logout',
                    ),
                ],
                flexibleSpace: Container(
                    decoration: BoxDecoration(
                        gradient: LinearGradient(
                            colors: [Colors.blue.shade900,
Colors.blue.shade700],
                            begin: Alignment.topCenter,
                            end: Alignment.bottomCenter,
                        ),
                    ),
                ),
            ),
            drawer: Drawer(
                child: Column(
                    children: [
                        DrawerHeader(
                            decoration: BoxDecoration(
                                gradient: LinearGradient(
                                    colors:
[Colors.blue.shade800, Colors.blue.shade600],
                                    begin: Alignment.topLeft,
                                    end:
Alignment.bottomRight,
                                ),
```

```
                            ),
                            child: Center(
                                child: Column(
                                    mainAxisAlignment:
MainAxisAlignment.center,
                                    children: [
                                        CircleAvatar(
                                            radius: 40,
                                            backgroundImage:
AssetImage('assets/splash_image.jpg'),
                                        ),
                                        const SizedBox(height:
12),
                                        const Text(
                                            'Project Name',
                                            style: TextStyle(
                                                color:
Colors.white,
                                                fontSize: 20,
                                                fontWeight:
FontWeight.bold,
                                            ),
                                        ),
                                    ],
                                ),
                            ),
                        ),
                        ListTile(
                            leading: const
Icon(Icons.category),
                            title: const Text('Categories'),
                            onTap: () {
                                Navigator.pop(context);
                                Navigator.pushNamed(context,
'/categories');
                            },
                        ),
                        ListTile(
                            leading: const
Icon(Icons.inventory),
                            title: const Text('All Products'),
                            onTap: () {
                                Navigator.pop(context);
                                Navigator.pushNamed(context,
'/products');
                            },
                        ),
                        const Divider(),
                        ListTile(
                            leading: const Icon(Icons.lock),
                            title: const Text('Change
```

```
Password'),
                                onTap: () {
                                    Navigator.pop(context);
                                    Navigator.push(
                                        context,
                                        MaterialPageRoute(
                                            builder: (context) =>
ChangePassword(),
                                        ),
                                    );
                                },
                            ),
                            const Spacer(),
                            ListTile(
                                leading: const Icon(Icons.logout),
                                title: const Text('Logout'),
                                onTap: () => _logout(context),
                            ),
                        ],
                    ),
                ),
                body: SingleChildScrollView(
                    child: Column(
                        children: [
                            SizedBox(height:
MediaQuery.of(context).padding.top + kToolbarHeight),
                            CarouselSlider(
                                options: CarouselOptions(
                                    height: 200.0,
                                    autoPlay: true,
                                    enlargeCenterPage: true,
                                    aspectRatio: 16/9,
                                    autoPlayCurve:
Curves.fastOutSlowIn,
                                    enableInfiniteScroll: true,
                                    autoPlayAnimationDuration:
const Duration(milliseconds: 800),
                                    viewportFraction: 0.8,
                                    onPageChanged: (index, reason)
{
                                        setState(() {
                                            _currentCarouselIndex
= index;
                                        });
                                    },
                                ),
                                items: carouselItems.map((item) {
                                    return Builder(
                                        builder: (BuildContext
context) {
                                            return Container(
```

```
                                                    width:
MediaQuery.of(context).size.width,
                                                    margin: const
EdgeInsets.symmetric(horizontal: 5.0),
                                                    decoration:
BoxDecoration(
                                                        borderRadius:
BorderRadius.circular(15),
                                                        image:
DecorationImage(
                                                            image:
AssetImage(item['image']),
                                                            fit:
BoxFit.cover,
                                                        ),
                                                    ),
                                                    child: Container(
                                                        decoration:
BoxDecoration(
                                                            gradient:
borderRadius: BorderRadius.circular(15),
LinearGradient(
                                                                begin:
Alignment.topCenter,
                                                                end:
Alignment.bottomCenter,
colors: [
Colors.transparent,
Colors.black.withOpacity(0.7),
                                                                ],
                                                            ),
                                                        ),
                                                        padding: const
EdgeInsets.all(20),
                                                        child: Column(
mainAxisAlignment: MainAxisAlignment.end,
crossAxisAlignment: CrossAxisAlignment.start,
                                                            children:
[
                                                                Text(
item['title'],
style: const TextStyle(
```

```
color: Colors.white,

fontSize: 20,

fontWeight: FontWeight.bold,
                                                                    ),
                                                                ),
                                                            ],
                                                        ),
                                                    ),
                                                );
                                            },
                                        );
                                    }).toList(),
                                ),
                                const SizedBox(height: 20),
                                Row(
                                    mainAxisAlignment:
MainAxisAlignment.center,
                                    children:
carouselItems.asMap().entries.map((entry) {
                                        return Container(
                                            width: 8.0,
                                            height: 8.0,
                                            margin: const
EdgeInsets.symmetric(vertical: 8.0, horizontal: 4.0),
                                            decoration: BoxDecoration(
                                                shape:
BoxShape.circle,
                                                color:
Theme.of(context).primaryColor.withOpacity(

_currentCarouselIndex == entry.key ? 0.9 : 0.4,
                                                ),
                                            ),
                                        );
                                    }).toList(),
                                ),
                                const SizedBox(height: 20),
                                GridView.builder(
                                    shrinkWrap: true,
                                    physics: const
NeverScrollableScrollPhysics(),
                                    padding: const EdgeInsets.all(16),
                                    gridDelegate: const
SliverGridDelegateWithFixedCrossAxisCount(
                                        crossAxisCount: 2,
                                        childAspectRatio: 1.1,
                                        crossAxisSpacing: 16,
                                        mainAxisSpacing: 16,
                                    ),
```

```
                            itemCount: gridItems.length,
                            itemBuilder: (context, index) {
                                final item = gridItems[index];
                                return Card(
                                    elevation: 4,
                                    shape:
RoundedRectangleBorder(
                                        borderRadius:
BorderRadius.circular(15),
                                    ),
                                    child: InkWell(
                                        onTap: () {
                                            // Handle grid
item tap
                                        },
                                        borderRadius:
BorderRadius.circular(15),
                                        child: Container(
                                            decoration:
BoxDecoration(
                                                borderRadius:
BorderRadius.circular(15),
                                                gradient:
LinearGradient(
                                                    colors: [

item['color'],

item['color'].withOpacity(0.8),
                                                    ],
                                                    begin:
Alignment.topLeft,
                                                    end:
Alignment.bottomRight,
                                                ),
                                            ),
                                            child: Column(

mainAxisAlignment: MainAxisAlignment.center,
                                                children: [
                                                    Icon(

item['icon'],
                                                        size:
40,
                                                        color:
Colors.white,
                                                    ),
                                                    const
SizedBox(height: 8),
                                                    Text(
```

```
item['title'],
                                                                    style:
const TextStyle(

color: Colors.white,

fontSize: 16,

fontWeight: FontWeight.bold,
                                                                        ),
                                                                    ),
                                                                ],
                                                            ),
                                                        ),
                                                    ),
                                                );
                                            },
                                        ),
                                    ],
                                ),
                            ),
                        );
                    }
}
```

ForgotPasswordScreen.dart

```
import 'package:flutter/material.dart'; // Adjust the path
import '../service/ForgotService.dart';
import 'LoginScreen.dart';

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

  @override
  State<ForgotPasswordScreen> createState() =>
_ForgotPasswordScreenState();
}

class _ForgotPasswordScreenState extends
State<ForgotPasswordScreen> {
  final _formKey = GlobalKey<FormState>();
  final _emailController = TextEditingController();
  bool _isLoading = false;

  Future<void> _resetPassword() async {
    if (_formKey.currentState!.validate()) {
      setState(() {
        _isLoading = true;
```

```
      });

      try {
        final result = await
AuthService.forgotPassword(_emailController.text);

        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text(result['message'])),
        );

        if (result['success']) {
          Navigator.pop(context);
        }
      } catch (e) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text('Error: ${e.toString()}')),
        );
      } finally {
        setState(() {
          _isLoading = false;
        });
      }
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Forgot Password', style:
TextStyle(color: Colors.white)),
        flexibleSpace: Container(
          decoration: BoxDecoration(
            gradient: LinearGradient(
              colors: [Colors.blue.shade900,
Colors.blue.shade700],
              begin: Alignment.topCenter,
              end: Alignment.bottomCenter,
            ),
          ),
        ),
      ),
      body:Align(
    alignment: Alignment.center,
    child:
      SingleChildScrollView(
        padding: const EdgeInsets.all(16.0),
        child: Form(
          key: _formKey,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.stretch,
```

```
            children: [
              Card(
                elevation: 5,
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(16),
                ),
                child: Padding(
                  padding: const EdgeInsets.all(16.0),
                  child: Column(
                    children: [
                      Image.asset(
                        'assets/forgot_image.jpg',
                        height: 250,
                        width: 450,
                      ),
                      SizedBox(height: 15,),
                      TextFormField(
                        controller: _emailController,
                        decoration: const InputDecoration(
                          labelText: 'Email',
                          border: OutlineInputBorder(),
                        ),
                        validator: (value) {
                          if (value == null || value.isEmpty)
{
                            return 'Please enter your email';
                          }
                          return null;
                        },
                      ),
                      const SizedBox(height: 24),
                      _isLoading
                          ? const Center(child:
CircularProgressIndicator())
                          : ElevatedButton(
                              onPressed: _resetPassword,
                              style: ElevatedButton.styleFrom(
                                backgroundColor:
Colors.blue.shade700,
                                shape: RoundedRectangleBorder(
                                  borderRadius:
BorderRadius.circular(8),
                                ),
                              ),
                              child: const Text('Reset
Password', style: TextStyle(color: Colors.white)),
                            ),
                    ],
                  ),
                ),
              ),
```

```
            ],
          ),
        ),
      ),
      ),
    );
  }
}
```

Services/RegisterService.dart

```
// lib/service/auth_service.dart
import 'dart:convert';
import 'package:http/http.dart' as http;
import '../constants/constants.dart';
class AuthService {

  static Future<Map<String, dynamic>> register({
    required String name,
    required String email,
    required String number,
    required String password,
  }) async {
    final response = await http.post(
      Uri.parse('${ApiConstants.baseUrl}/register.php'),
      body: {
        'name': name,
        'email': email,
        'number': number,
        'password': password,
        'token':
DateTime.now().millisecondsSinceEpoch.toString(),
      },
    );

    return json.decode(response.body);
  }
}
```

LoginService.dart

```
// service/auth_service.dart
import 'dart:convert';
import 'package:http/http.dart' as http;
import '../constants/constants.dart';
class AuthService {
```

```
  static Future<Map<String, dynamic>> login(String email,
String password) async {
    final response = await http.post(
      Uri.parse('${ApiConstants.baseUrl}/login.php'),
      body: {
        'email': email,
        'password': password,
      },
    );

    if (response.statusCode == 200) {
      return json.decode(response.body);
    } else {
      throw Exception('Login failed with status:
${response.statusCode}');
    }
  }
}
```

ForgotService.dart

```
// service/auth_service.dart
import 'dart:convert';
import 'package:http/http.dart' as http;
import '../constants/constants.dart';
class AuthService {
  static const String _forgotPasswordUrl =
'${ApiConstants.baseUrl}/forgot.php';

  static Future<Map<String, dynamic>> forgotPassword(String
email) async {
    try {
      final response = await http.post(
        Uri.parse(_forgotPasswordUrl),
        body: {
          'email': email,
        },
      );

      if (response.statusCode == 200) {
        return json.decode(response.body);
      } else {
        return {
          'success': false,
          'message': 'Server error. Please try again later.',
        };
      }
    } catch (e) {
```

```
      return {
        'success': false,
        'message': 'An error occurred. Please try again.',
      };
    }
  }
}
```