EXPERIMENT NO :- 03

AIM: - To include icons, images, fonts in Flutter app.

Theory: -

Flutter is a versatile open-source UI framework , which allows developers to build natively compiled applications for mobile, web, and desktop platforms from a single codebase. One of the key strengths of Flutter is its flexibility in creating highly customizable UIs. This practical focuses on incorporating essential visual elements—icons, images, and custom fonts—into a Flutter application. These elements enhance the visual appeal and usability of the app, providing an engaging experience for users.
A flutter app when built has both assets (resources) and code. Assets are available and deployed during runtime. The asset is a file that can include static data, configuration files, icons, and images. The Flutter app supports many image formats, such as JPEG, WebP, PNG, GIF, animated WebP/GIF, BMP, and WBMP.
Visual elements play a significant role in app development.

- Enhanced User Experience: Images and icons make your app visually appealing and user-friendly.
- Information Conveyance: They convey information quickly and intuitively. A well-chosen icon can replace lengthy text.
- Branding: Custom icons and images reinforce your app's branding, making it memorable.

Adding Icons in Flutter

Flutter provides built-in material design icons through the Icons class. Custom icons can also be added using third-party packages such as flutter_launcher_icons and font_awesome_flutter.

```
Icon(
Icons.home,
size: 40,
);
```

Adding Images in Flutter

Flutter supports images from three sources:

1.Assets (Stored locally in the project)

- Place the image inside the assets/images folder in the project.
  Declare the image in pubspec.yaml

```
flutter:
assets:
- assets/images/sample.png
```

- Display the image in the app

```
Image.asset('assets/images/sample.png');
```

2.Network (Fetched from the internet)

- Displaying images from the internet or network is very simple. Flutter provides a built-in method Image.network to work with images from a URL. The Image.network method also allows you to use some optional properties, such as height, width, color, fit, and many more.

```
Image.network('https://example.com/sample.jpg');
```

3.Memory or File (Stored on the device)

Adding Custom Fonts in Flutter

By default, Flutter uses the Roboto font, but custom fonts can be added for a unique UI.

- Download the font and place it in the assets/fonts/ folder.
- Declare the font in pubspec.yaml
- Use the font in the app

```
Text(
'Custom Font Example',
style: TextStyle(fontFamily: 'CustomFont', fontSize: 24),
);
```

Code: -

login_screen.dart

```
import 'package:flutter/material.dart';
import
'package:firebase_auth/firebase_auth.dart';
import 'main_screen.dart'; // Import the
MainScreen (not HomeScreen)

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

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

class _LoginScreenState extends
State<LoginScreen> {
  // Controllers for input fields
  final TextEditingController emailController =
TextEditingController();
  final TextEditingController
passwordController = TextEditingController();

  // Firebase Auth instance
  final FirebaseAuth _auth =
FirebaseAuth.instance;

  bool _isLoading = false;

  @override
  void dispose() {
    emailController.dispose();
    passwordController.dispose();
    super.dispose();
  }

  /// Sign in using Firebase Authentication
  Future<void> _login() async {
    setState(() {
      _isLoading = true;
    });

    try {
      // Attempt sign in with email and password
      await
_auth.signInWithEmailAndPassword(
        email: emailController.text.trim(),
        password: passwordController.text,
      );

      // On successful login, navigate to the
MainScreen with BottomNavigationBar.
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) =>
const MainScreen()), // Navigate to
MainScreen
      );
    } on FirebaseAuthException catch (e) {
      String message = 'Login failed';
      if (e.code == 'user-not-found') {
        message = 'No user found for that
email.';
      } else if (e.code == 'wrong-password') {
        message = 'Incorrect password
provided.';
      }

ScaffoldMessenger.of(context).showSnackBar
(
        SnackBar(content: Text(message)),
      );
    } catch (e) {

ScaffoldMessenger.of(context).showSnackBar
(
        const SnackBar(content: Text('An error
occurred.')),
      );
    }

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.black,
      appBar: AppBar(
        backgroundColor: Colors.black,
        title: const Text("Log In", style:
TextStyle(color: Colors.white)),
        centerTitle: true,
        actions: [
          // Skip button for direct access (login
later)
          TextButton(
            onPressed: () {
              Navigator.pushReplacement(
                context,
                MaterialPageRoute(builder:
(context) => const MainScreen()), // Navigate
to MainScreen
              );
            },
            child: const Text("Skip", style:
TextStyle(color: Colors.white, fontSize: 16)),
          ),
        ],
        leading: IconButton(
          icon: const Icon(Icons.arrow_back,
color: Colors.white),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
      ),
      body: Padding(
        padding: const
EdgeInsets.symmetric(horizontal: 20),
        child: Column(
          children: [
            const SizedBox(height: 30),
            // Email TextField
            TextField(
              controller: emailController,
              style: const TextStyle(color:
Colors.white),
              decoration: InputDecoration(
                labelText: "Email Address",
                labelStyle: const TextStyle(color:
Colors.white70),
                enabledBorder: OutlineInputBorder(
                  borderSide: const
BorderSide(color: Colors.white38),
                  borderRadius:
BorderRadius.circular(8),
                ),
                focusedBorder: OutlineInputBorder(
                  borderSide: const
BorderSide(color: Colors.blue),
                  borderRadius:
BorderRadius.circular(8),
                ),
              ),
            ),
            const SizedBox(height: 20),
            // Password TextField
            TextField(
              controller: passwordController,
              obscureText: true,
              style: const TextStyle(color:
Colors.white),
              decoration: InputDecoration(
                labelText: "Password",
                labelStyle: const TextStyle(color:
Colors.white70),
                enabledBorder: OutlineInputBorder(
                  borderSide: const
BorderSide(color: Colors.white38),
                  borderRadius:
BorderRadius.circular(8),
                ),
                focusedBorder: OutlineInputBorder(
                  borderSide: const
BorderSide(color: Colors.blue),
                  borderRadius:
BorderRadius.circular(8),
                ),
              ),
            ),
            const SizedBox(height: 20),
            // Login button or a loading indicator
            _isLoading
                ? const CircularProgressIndicator()
                : ElevatedButton(
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.blue,
                minimumSize: const
Size(double.infinity, 50),
              ),
              onPressed: _login,
              child: const Text("Log In", style:
TextStyle(color: Colors.white)),
            ),
            const SizedBox(height: 10),
            TextButton(
              onPressed: () {
                // Implement forgot password logic if
desired.
              },
              child: const Text("Forgot password?",
style: TextStyle(color: Colors.blue)),
            ),
            const SizedBox(height: 20),
            const Row(
              children: [
                Expanded(child: Divider(color:
Colors.white)),
                Padding(
                  padding:
EdgeInsets.symmetric(horizontal: 10),
                  child: Text("OR", style:
TextStyle(color: Colors.white)),
                ),
                Expanded(child: Divider(color:
Colors.white)),
              ],
            ),
            // Additional social login buttons can
be added here
            const Spacer(),
            // Navigation to registration screen
            TextButton(
              onPressed: () {
                Navigator.pushNamed(context,
'/register');
              },
              child: const Text("Don't have an
account? Register", style: TextStyle(color:
Colors.blue, fontSize: 16)),
            ),
            const SizedBox(height: 20),
          ],
        ),
      ),
    );
  }
}
```

registration_screen.dart

```
import 'package:flutter/material.dart';
import
'package:firebase_auth/firebase_auth.dart';

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

  @override
  State<RegistrationScreen> createState() =>
_RegistrationScreenState();
}

class _RegistrationScreenState extends
State<RegistrationScreen> {
  // Controllers for input fields
  final TextEditingController nameController =
TextEditingController();
  final TextEditingController emailController =
TextEditingController();
  final TextEditingController
passwordController = TextEditingController();
  final TextEditingController
confirmPasswordController =
TextEditingController();

  // Firebase Auth instance
  final FirebaseAuth _auth =
FirebaseAuth.instance;

  bool _isLoading = false;

  @override
  void dispose() {
    nameController.dispose();
    emailController.dispose();
    passwordController.dispose();
    confirmPasswordController.dispose();
    super.dispose();
  }

  /// Register a new user with Firebase
Authentication
  Future<void> _register() async {
    final email = emailController.text.trim();
    final password = passwordController.text;
    final confirmPassword =
confirmPasswordController.text;

    if (password != confirmPassword) {

ScaffoldMessenger.of(context).showSnackBar
(
        const SnackBar(content:
Text("Passwords do not match")),
      );
      return;
    }

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

    try {
      // Create the user
      UserCredential userCredential = await
_auth.createUserWithEmailAndPassword(
        email: email,
        password: password,
      );

      // Optionally update the display name
      if (userCredential.user != null &&
nameController.text.trim().isNotEmpty) {
        await
userCredential.user!.updateDisplayName(na
meController.text.trim());
      }

      // On successful registration, navigate to
main screen.

Navigator.pushReplacementNamed(context,
'/main');
    } on FirebaseAuthException catch (e) {
      String message = 'Registration failed';
      if (e.code == 'weak-password') {
        message = 'The password provided is
too weak.';
      } else if (e.code == 'email-already-in-use')
{
        message = 'An account already exists for
that email.';
      }

ScaffoldMessenger.of(context).showSnackBar
(
        SnackBar(content: Text(message)),
      );
    } catch (e) {

ScaffoldMessenger.of(context).showSnackBar
(
    const SnackBar(content: Text("An error
occurred")),
   );
  }

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

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   backgroundColor: Colors.black,
   appBar: AppBar(
    backgroundColor: Colors.black,
    title: const Text("Register", style:
TextStyle(color: Colors.white)),
    centerTitle: true,
    leading: IconButton(
     icon: const Icon(Icons.arrow_back,
color: Colors.white),
     onPressed: () {
      Navigator.pop(context);
     },
    ),
   ),
   body: SingleChildScrollView(
    padding: const
EdgeInsets.symmetric(horizontal: 20, vertical:
20),
    child: Column(
     children: [
      const SizedBox(height: 20),
      // Full Name field (optional)
      TextField(
       controller: nameController,
       style: const TextStyle(color:
Colors.white),
       decoration: InputDecoration(
        labelText: "Full Name",
        labelStyle: const TextStyle(color:
Colors.white70),
        enabledBorder: OutlineInputBorder(
         borderSide: const
BorderSide(color: Colors.white38),
         borderRadius:
BorderRadius.circular(8),
        ),
        focusedBorder: OutlineInputBorder(
         borderSide: const
BorderSide(color: Colors.blue),
         borderRadius:
BorderRadius.circular(8),
        ),
       ),
      ),
      const SizedBox(height: 20),
      // Email Address field
      TextField(
       controller: emailController,
       style: const TextStyle(color:
Colors.white),
       decoration: InputDecoration(
        labelText: "Email Address",
        labelStyle: const TextStyle(color:
Colors.white70),
        enabledBorder: OutlineInputBorder(
         borderSide: const
BorderSide(color: Colors.white38),
         borderRadius:
BorderRadius.circular(8),
        ),
        focusedBorder: OutlineInputBorder(
         borderSide: const
BorderSide(color: Colors.blue),
         borderRadius:
BorderRadius.circular(8),
        ),
       ),
      ),
      const SizedBox(height: 20),
      // Password field
      TextField(
       controller: passwordController,
       obscureText: true,
       style: const TextStyle(color:
Colors.white),
       decoration: InputDecoration(
        labelText: "Password",
        labelStyle: const TextStyle(color:
Colors.white70),
        enabledBorder: OutlineInputBorder(
         borderSide: const
BorderSide(color: Colors.white38),
         borderRadius:
BorderRadius.circular(8),
        ),
        focusedBorder: OutlineInputBorder(
         borderSide: const
BorderSide(color: Colors.blue),
         borderRadius:
BorderRadius.circular(8),
        ),
       ),
      ),
      const SizedBox(height: 20),
      // Confirm Password field
      TextField(
       controller:
confirmPasswordController,
       obscureText: true,
       style: const TextStyle(color:
Colors.white),
       decoration: InputDecoration(
        labelText: "Confirm Password",
        labelStyle: const TextStyle(color:
Colors.white70),
        enabledBorder: OutlineInputBorder(
         borderSide: const
BorderSide(color: Colors.white38),
         borderRadius:
BorderRadius.circular(8),
        ),
        focusedBorder: OutlineInputBorder(
         borderSide: const
BorderSide(color: Colors.blue),
         borderRadius:
BorderRadius.circular(8),
import 'package:flutter/material.dart';

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

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

class _HomeScreenState extends
State<HomeScreen> {
 int _currentCarouselIndex = 0;
 final PageController _pageController =
PageController();

 final List<Map<String, dynamic>>
carouselData = [
```

```
        ),
       ),
      ),
      const SizedBox(height: 20),
      _isLoading
        ? const CircularProgressIndicator()
        : ElevatedButton(
       style: ElevatedButton.styleFrom(
        backgroundColor: Colors.blue,
        minimumSize: const
Size(double.infinity, 50),
       ),
       onPressed: _register,
       child: const Text("Register", style:
TextStyle(color: Colors.white)),
      ),
      const SizedBox(height: 10),
      // Optionally add a link to the login
page if already have an account.
     ],
    ),
   ),
  );
 }
}
```

home_screen.dart

```
  {
   'icon': Icons.fitness_center,
   'title': 'Featured Workout',
   'description': 'Try our new HIIT routine for
maximum burn!',
   'color': Colors.blueAccent,
  },
  {
   'icon': Icons.local_dining,
   'title': 'Nutrition Tip',
   'description': 'Include more protein in your
meals for muscle gain.',
   'color': Colors.green,
  },
  {
   'icon': Icons.self_improvement,
   'title': 'Mindfulness',
      'description': 'Take a moment to meditate
and destress.',
      'color': Colors.purple,
    },
  ];

  @override
  void dispose() {
    _pageController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Home"),
        backgroundColor: Colors.black,
        centerTitle: true,
      ),
      backgroundColor: Colors.black,
      body: SingleChildScrollView(
        child: Column(
          children: [
            // Carousel Section with Custom
Widgets
            SizedBox(
              height: 200,
              child: PageView.builder(
                controller: _pageController,
                itemCount: carouselData.length,
                onPageChanged: (index) {
                  setState(() {
                    _currentCarouselIndex = index;
                  });
                },
                itemBuilder: (context, index) {
                  final item = carouselData[index];
                  return Container(
                    margin: const
EdgeInsets.symmetric(horizontal: 10, vertical:
10),
                    decoration: BoxDecoration(
                      borderRadius:
BorderRadius.circular(12),
                      color:
item['color'].withOpacity(0.2),
                    ),
                    padding: const
EdgeInsets.all(16),
                    child: Row(
                      children: [
                        Icon(
                          item['icon'],
                          size: 48,
                          color: item['color'],
                        ),
                        const SizedBox(width: 16),
                        Expanded(
                          child: Column(
                            mainAxisAlignment:
MainAxisAlignment.center,
                            crossAxisAlignment:
CrossAxisAlignment.start,
                            children: [
                              Text(
                                item['title'],
                                style: const TextStyle(
                                  color: Colors.white,
                                  fontSize: 20,
                                  fontWeight:
FontWeight.bold,
                                ),
                              ),
                              const SizedBox(height: 8),
                              Text(
                                item['description'],
                                style: const TextStyle(
                                  color: Colors.white70,
                                  fontSize: 16,
                                ),
                              ),
                            ],
                          ),
                        ),
                      ],
                    ),
                  );
                },
              ),
            ),
            // Carousel Indicator Dots
            Row(
              mainAxisAlignment:
MainAxisAlignment.center,
              children:
List.generate(carouselData.length, (index) {
```

```
                return AnimatedContainer(
                  duration: const Duration(milliseconds: 300),
                  margin: const EdgeInsets.symmetric(horizontal: 4),
                  height: 8,
                  width: _currentCarouselIndex == index ? 20 : 8,
                  decoration: BoxDecoration(
                    color: _currentCarouselIndex == index ? Colors.blue : Colors.white54,
                    borderRadius: BorderRadius.circular(4),
                  ),
                );
              }),
            ),
            const SizedBox(height: 20),

            // Trackers Section
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 16),
              child: Row(
                children: const [
                  Text(
                    "Trackers",
                    style: TextStyle(
                      color: Colors.white,
                      fontSize: 20,
                      fontWeight: FontWeight.bold,
                    ),
                  ),
                ],
              ),
            ),
            const SizedBox(height: 10),
            SizedBox(
              height: 150,
              child: ListView(
                scrollDirection: Axis.horizontal,
                padding: const EdgeInsets.symmetric(horizontal: 16),
                children: [
                  _buildTrackerCard(
                    icon: Icons.favorite,
                    title: "Heart Rate",
                    value: "72 bpm",
                  ),
                  const SizedBox(width: 10),
                  _buildTrackerCard(
                    icon: Icons.directions_run,
                    title: "Steps",
                    value: "5,432",
                  ),
                  const SizedBox(width: 10),
                  _buildTrackerCard(
                    icon: Icons.local_fire_department,
                    title: "Calories",
                    value: "1,234 cal",
                  ),
                ],
              ),
            ),
            const SizedBox(height: 20),

            // Goals Section
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 16),
              child: Row(
                children: const [
                  Text(
                    "Your Goals",
                    style: TextStyle(
                      color: Colors.white,
                      fontSize: 20,
                      fontWeight: FontWeight.bold,
                    ),
                  ),
                ],
              ),
            ),
            const SizedBox(height: 10),
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 16),
              child: Row(
                children: [
                  Expanded(
                    child: _buildGoalCard(
                      title: "Remaining Calories",
                      value: "850 cal",
                      icon: Icons.local_fire_department,
                      color: Colors.orange,
                    ),
                  ),
                  const SizedBox(width: 10),
                  Expanded(
                    child: _buildGoalCard(
                      title: "Water Intake",
                      value: "5/8 glasses",
                      icon: Icons.water,
                      color: Colors.lightBlue,
                    ),
                  ),
                ],
              ),
            ),
            const SizedBox(height: 20),

            // Graph Section
            Padding(
              padding: const
EdgeInsets.symmetric(horizontal: 16),
              child: Row(
                children: const [
                  Text(
                    "Progress Graph",
                    style: TextStyle(
                      color: Colors.white,
                      fontSize: 20,
                      fontWeight: FontWeight.bold,
                    ),
                  ),
                ],
              ),
            ),
            const SizedBox(height: 10),
            _buildGraphWidget(),
            const SizedBox(height: 30),
          ],
        ),
      ),
    );
  }

  // Tracker Card
  Widget _buildTrackerCard({
    required IconData icon,
    required String title,
    required String value,
  }) {
    return Container(
      width: 140,
      padding: const EdgeInsets.all(16),
      decoration: BoxDecoration(
        color: Colors.grey[900],
        borderRadius: BorderRadius.circular(12),
      ),
      child: Column(
        mainAxisAlignment:
MainAxisAlignment.center,
        children: [
          Icon(icon, size: 40, color: Colors.blue),
          const SizedBox(height: 10),
          Text(
            title,
            style: const TextStyle(
              color: Colors.white70,
              fontSize: 16,
            ),
          ),
          const SizedBox(height: 5),
          Text(
            value,
            style: const TextStyle(
              color: Colors.white,
              fontSize: 18,
              fontWeight: FontWeight.bold,
            ),
          )
        ],
      ),
    );
  }

  // Goal Card
  Widget _buildGoalCard({
    required String title,
    required String value,
    required IconData icon,
    required Color color,
  }) {
    return Container(
      padding: const EdgeInsets.all(16),
      decoration: BoxDecoration(
        color: Colors.grey[900],
        borderRadius: BorderRadius.circular(12),
      ),
      child: Column(
        children: [
          Icon(icon, size: 40, color: color),
          const SizedBox(height: 10),
          Text(
          title,
          textAlign: TextAlign.center,
          style: const TextStyle(
            color: Colors.white70,
            fontSize: 14,
          ),
        ),
        const SizedBox(height: 5),
        Text(
          value,
          style: const TextStyle(
            color: Colors.white,
            fontSize: 16,
            fontWeight: FontWeight.bold,
          ),
        )
      ],
    ),
  );
}

// Placeholder for Graph Widget
```

```
  Widget _buildGraphWidget() {
    return Container(
      margin: const
EdgeInsets.symmetric(horizontal: 16),
      height: 200,
      decoration: BoxDecoration(
        color: Colors.grey[900],
        borderRadius: BorderRadius.circular(12),
      ),
      child: const Center(
        child: Text(
          "Graph \n(Progress Over Time)",
          textAlign: TextAlign.center,
          style: TextStyle(
            color: Colors.white70,
            fontSize: 16,
          ),
        ),
      ),
    );
  }
}
```

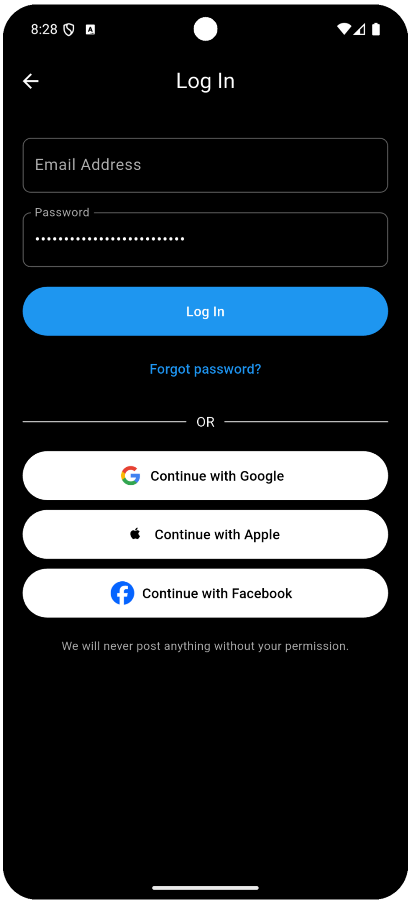
main_screen.dart

```
import 'package:flutter/material.dart';
import 'home_screen.dart';
import 'log_food_screen.dart';
import 'settings_screen.dart';

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

  @override
  State<MainScreen> createState() =>
_MainScreenState();
}

class _MainScreenState extends
State<MainScreen> {
  int _currentIndex = 0;

  // List of pages corresponding to each
bottom nav item.
  final List<Widget> _pages = const [
    HomeScreen(),
    LogFoodScreen(),
    SettingsScreen(),
```

```
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.black, // Ensures
a black background for the screen
      body: _pages[_currentIndex], // Display
selected page
      bottomNavigationBar:
BottomNavigationBar(
        backgroundColor: Colors.grey[850], //
Ensure a distinct background color
        currentIndex: _currentIndex,
        unselectedItemColor: Colors.white70,
        selectedItemColor: Colors.blue,
        showUnselectedLabels: true,
        items: const [
          BottomNavigationBarItem(
            icon: Icon(Icons.home),
            label: "Home",
          ),
          BottomNavigationBarItem(
    icon: Icon(Icons.fastfood),
    label: "Log Food",
  ),
  BottomNavigationBarItem(
    icon: Icon(Icons.settings),
    label: "Settings",
  ),
],
onTap: (index) {
          setState(() {
            _currentIndex = index; // Update the
current selected index
          });
        },
      ),
    );
  }
}
```

Shreyash G Dhekane D15A Roll No:16

Screenshots:


8:28

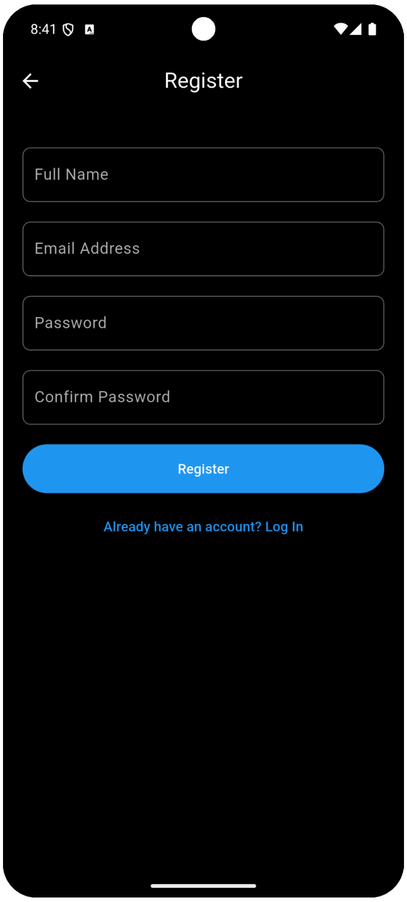
Log In

Email Address

Password

••••••••••••••••••••••••••••

We will never post anything without your permission.


8:41

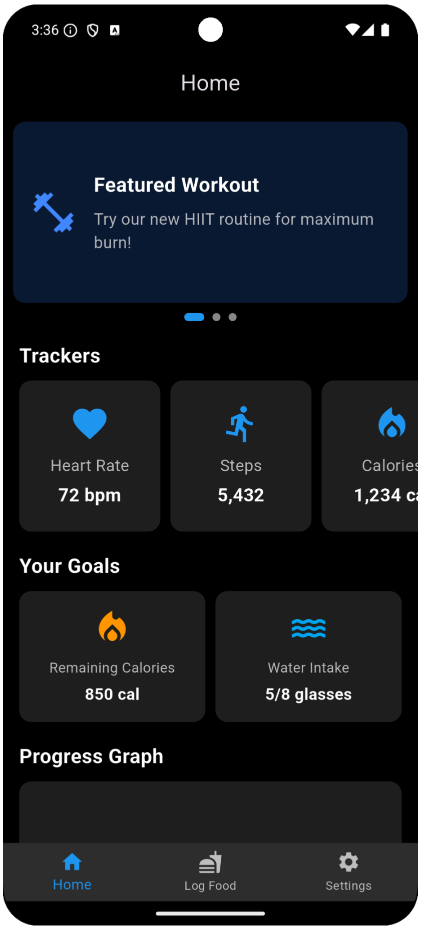
Register

Full Name

Email Address

Password

Confirm Password


3:36

Home

Featured Workout

Try our new HIIT routine for maximum burn!

Trackers

Heart Rate 72 bpm

Steps 5,432

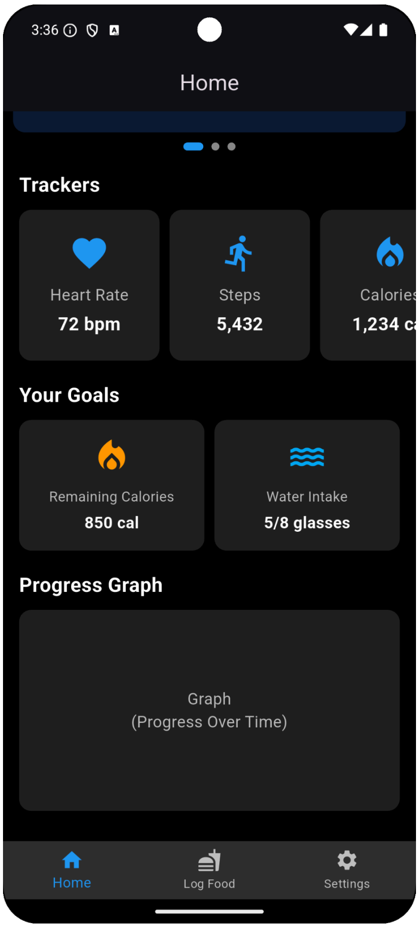
Calories 1,234 ca

Your Goals

Remaining Calories 850 cal

Water Intake 5/8 glasses

Progress Graph


3:36

Home

Trackers

Heart Rate 72 bpm

Steps 5,432

Calories 1,234 ca

Your Goals

Remaining Calories 850 cal

Water Intake 5/8 glasses

Progress Graph

Graph
(Progress Over Time)