





main.dart

```
import 'package:flutter/material.dart';
import 'package:api_locate/home.dart';
import 'package:api_locate/BMI.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: MyPage(),
    );
  }
}

class MyPage extends StatefulWidget {
  @override
  _MyPageState createState() => _MyPageState();
}

class _MyPageState extends State<MyPage> {
  int _selectedIndex = 0;
  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }
}
```

```

    });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: IndexedStack(
      index: _selectedIndex,
      children: [
        Home(),
        BMI(),
      ],
    ),
    bottomNavigationBar: BottomNavigationBar(
      items: const <BottomNavigationBarItem>[
        BottomNavigationBarItem(
          icon: Icon(Icons.home, size: 30),
          label: 'FOOD RECIPES',
        ),
        BottomNavigationBarItem(
          icon: Icon(Icons.scale, size: 30),
          label: 'BMI CALCULATOR',
        ),
      ],
      currentIndex: _selectedIndex,
      selectedItemColor: Colors.white,
      unselectedItemColor: Colors.white54,
      backgroundColor: Color(0xFFfe5d26),
      onTap: _onItemTapped,
      selectedLabelStyle: TextStyle(
        fontSize: 20,
        fontWeight: FontWeight.bold,
      ),
      unselectedLabelStyle: TextStyle(
        fontSize: 20,
        fontWeight: FontWeight.bold,
      ),
    ),
  );
}
}

```

home.dart

```

import 'package:flutter/material.dart';
import 'package:api_locate/recipe.dart';

class Home extends StatefulWidget {
  const Home({Key? key}) : super(key: key);

  @override
  _HomeState createState() => _HomeState();
}

class _HomeState extends State<Home> {
  @override
  Widget build(BuildContext context) {

```



```

        SizedBox(height: 40),
        SizedBox(
          height: 85,
          child: Container(
            decoration: BoxDecoration(
              color: Color(0xFF7EBC89),
              border: Border.all(color: Colors.black, width: 2.5),
              borderRadius: BorderRadius.circular(20),
            ),
            child: ListTile(
              title: Text(
                'BiteSmart & Fit.',
                style: TextStyle(
                  fontSize: 25,
                  fontWeight: FontWeight.bold,
                ),
              ),
              subtitle: Text(
                'Your favorite app to keep you on track.',
                style: TextStyle(
                  fontSize: 16,
                  fontWeight: FontWeight.bold,
                ),
              ),
            ),
          ),
        ),
        SizedBox(height: 20),
        Text("Features"),
        SizedBox(height: 20),
        SizedBox(
          height: 85,
          child: Container(
            decoration: BoxDecoration(
              color: Color(0xFFFAEDCA),
              border: Border.all(color: Colors.black, width: 2.5),
              borderRadius: BorderRadius.circular(20),
            ),
            child: Row(
              children: [
                SizedBox(width: 25),
                Icon(
                  Icons.search,
                  size: 40,
                  color: Colors.black,
                ),
                SizedBox(width: 20),
                Text(
                  'Search Recipe',
                  style: TextStyle(
                    fontSize: 20,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                SizedBox(width: 20),
                SizedBox(
                  width: 100,

```



```

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          SizedBox(height: 40),
          Text(
            'HEIGHT',
            style: TextStyle(
              fontSize: 35,
              fontWeight: FontWeight.bold,
            ),
          ),
          Expanded(
            child: Container(
              decoration: BoxDecoration(
                color: Color(0xFFFAEDCA),
                border: Border.all(color: Colors.black, width: 2.5),
                borderRadius: BorderRadius.circular(20),
              ),
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  Column(
                    children: [
                      Text(
                        '${height.toStringAsFixed(2)} CM',
                        style: TextStyle(
                          fontSize: 20,
                          fontWeight: FontWeight.bold,
                        ),
                      ),
                      SizedBox(width: 20),
                      SliderTheme(
                        data: SliderTheme.of(context).copyWith(
                          activeTrackColor: Color(0xFFfe5d26),
                          inactiveTrackColor: Colors.grey,
                          trackShape: RectangularSliderTrackShape(),
                          trackHeight: 6.0, // Change the track height
                          thumbColor: Color(0xFFfe5d26), // Change the
thumb color
                          thumbShape:
RoundSliderThumbShape(enabledThumbRadius: 12.0), // Change the thumb shape
                          overlayColor: Color(0xFFfe5d26).withOpacity(0.3),
// Change the overlay color
                          overlayShape:
RoundSliderOverlayShape(overlayRadius: 20.0), // Change the overlay shape
                        ),
                        child: Slider(
                          value: height,
                          min: 100,
                          max: 220,

```


[illegible]

[illegible]

```

        );
      },
      child: Text('CALCULATE'),
    ),
  ],
),
),
);
}

double calculateBMI(double height, double weight) {
  double heightInMeter = height / 100;
  return weight / (heightInMeter * heightInMeter);
}

String getBMIResult(double bmi) {
  if (bmi < 18.5) {
    return 'Underweight';
  } else if (bmi >= 18.5 && bmi < 25) {
    return 'Normal weight';
  } else if (bmi >= 25 && bmi < 30) {
    return 'Overweight';
  } else {
    return 'Obese';
  }
}
}

```

recipe.dart

```

import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

class RecipeSearchPage extends StatefulWidget {
  @override
  _RecipeSearchPageState createState() => _RecipeSearchPageState();
}

class _RecipeSearchPageState extends State<RecipeSearchPage> {
  TextEditingController _searchController = TextEditingController();
  List<dynamic> _recipes = [];

  Future<void> _searchRecipes(String query) async {
    final String apiKey = 'd293941af3d8419d97ed8449f7c0c83b';
    final String apiUrl =
'https://api.spoonacular.com/recipes/complexSearch';

    final response = await http.get(
      Uri.parse('$apiUrl?number=20&query=$query&apiKey=$apiKey'),
      headers: {'Content-Type': 'application/json'},
    );

    if (response.statusCode == 200) {
      setState(() {
        _recipes = json.decode(response.body)['results'];
      });
    } else {

```

```

        // Handle errors
        print('Error: ${response.statusCode}');
    }
}

Future<void> _showIngredients(int recipeId) async {
    final String apiKey = 'd293941af3d8419d97ed8449f7c0c83b';
    final String apiUrl =
'https://api.spoonacular.com/recipes/$recipeId/information';

    final response = await http.get(
        Uri.parse('$apiUrl?apiKey=$apiKey'),
        headers: {'Content-Type': 'application/json'},
    );

    if (response.statusCode == 200) {
        final Map<String, dynamic> recipeDetails = json.decode(response.body);

        showDialog(
            context: context,
            builder: (BuildContext context) {
                return AlertDialog(
                    title: Text('Ingredients'),
                    content: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                            for (var ingredient in recipeDetails['extendedIngredients'])
                                Text('- ${ingredient['original']}'),
                        ],
                    ),
                    actions: [
                        TextButton(
                            onPressed: () {
                                Navigator.of(context).pop();
                            },
                            child: Text('Close'),
                        ),
                    ],
                );
            },
        );
    } else {
        // Handle errors
        print('Error: ${response.statusCode}');
    }
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text(
                '\t\t\t\t\tSEARCH RECIPE ID',
                style: TextStyle(
                    fontSize: 20,
                    fontWeight: FontWeight.bold,
                ),
            ),
        ),
    );
}

```

```

    ),
    backgroundColor: Color(0xFFfe5d26),
  ),
  body: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      children: [
        TextField(
          controller: _searchController,
          decoration: InputDecoration(
            labelText: 'ENTER FOOD (EX. CHICKEN)',
            suffixIcon: IconButton(
              icon: Icon(Icons.search),
              onPressed: () {
                _searchRecipes(_searchController.text);
              },
            ),
          ),
        ),
        SizedBox(height: 16),
        Expanded(
          child: _recipes.isNotEmpty
            ? ListView.builder(
                itemCount: _recipes.length,
                itemBuilder: (context, index) {
                  return ListTile(
                    title: Text(
                      'Dish: ${_recipes[index]['title']}',
                    ),
                    onTap: () {
                      _showIngredients(_recipes[index]['id']);
                    },
                  );
                },
              )
            : Center(
                child: Text('No recipes found.'),
              ),
        ),
      ],
    ),
  );
}

```