**NON FORMAL-NFC MAD USING FLUTTER**

**PROJECT NAME: LOST AND FOUND APP**

CODE:-

import 'package:flutter/material.dart';

import 'package:firebase\_core/firebase\_core.dart';

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'firebase\_options.dart';

import 'package:intl/intl.dart';

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp(

options: DefaultFirebaseOptions.currentPlatform, // Works for Web + Mobile

);

runApp(const CalorieApp());

}

class CalorieApp extends StatelessWidget {

const CalorieApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Calorie Meter',

theme: ThemeData(

primarySwatch: Colors.green,

scaffoldBackgroundColor: Colors.grey[100],

),

home: const HomePage(),

);

}

}

class HomePage extends StatelessWidget {

const HomePage({super.key});

@override

Widget build(BuildContext context) {

final String today = DateFormat('yyyy-MM-dd').format(DateTime.now());

return Scaffold(

appBar: AppBar(

title: const Text("Calorie Meter"),

backgroundColor: Colors.green.shade700,

),

body: Container(

decoration: const BoxDecoration(

gradient: LinearGradient(

colors: [Colors.lightGreen, Colors.greenAccent],

begin: Alignment.topLeft,

end: Alignment.bottomRight,

),

),

child: StreamBuilder<QuerySnapshot>(

stream: FirebaseFirestore.instance

.collection("dailyMeals")

.where("date", isEqualTo: today)

.snapshots(),

builder: (context, snapshot) {

if (snapshot.hasError) {

return Center(

child: Text(

"Error: ${snapshot.error}",

style: const TextStyle(color: Colors.red, fontSize: 16),

),

);

}

if (!snapshot.hasData) {

return const Center(child: CircularProgressIndicator());

}

final docs = snapshot.data!.docs;

int totalCalories = 0;

for (var doc in docs) {

final data = doc.data() as Map<String, dynamic>;

totalCalories += int.tryParse(data["calories"].toString()) ?? 0;

}

return Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const Text(

"Today's Calories",

style: TextStyle(

fontSize: 26,

fontWeight: FontWeight.bold,

color: Colors.white,

),

),

const SizedBox(height: 15),

Container(

padding: const EdgeInsets.all(30),

decoration: BoxDecoration(

color: Colors.white,

shape: BoxShape.circle,

boxShadow: [

BoxShadow(

color: Colors.black26,

blurRadius: 12,

offset: Offset(0, 6),

),

],

),

child: Text(

"$totalCalories kcal",

style: const TextStyle(

fontSize: 42,

fontWeight: FontWeight.bold,

color: Colors.green,

),

),

),

const SizedBox(height: 40),

ElevatedButton.icon(

style: ElevatedButton.styleFrom(

backgroundColor: Colors.orange,

padding: const EdgeInsets.symmetric(

horizontal: 30, vertical: 15),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(20)),

),

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(builder: (\_) => const AddMealPage()),

);

},

icon: const Icon(Icons.add, size: 24, color: Colors.white),

label: const Text("Add Meal",

style: TextStyle(fontSize: 18, color: Colors.white)),

),

const SizedBox(height: 15),

TextButton(

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(builder: (\_) => const ViewMealsPage()),

);

},

child: const Text(

"View All Meals",

style: TextStyle(color: Colors.white, fontSize: 16),

),

)

],

),

);

},

),

),

);

}

}

class AddMealPage extends StatefulWidget {

const AddMealPage({super.key});

@override

State<AddMealPage> createState() => \_AddMealPageState();

}

class \_AddMealPageState extends State<AddMealPage> {

final \_foodController = TextEditingController();

Future<void> \_addMeal() async {

final foodName = \_foodController.text.trim().toLowerCase();

final String today = DateFormat('yyyy-MM-dd').format(DateTime.now());

if (foodName.isEmpty) {

\_show("Enter food name");

return;

}

try {

final foodDoc = await FirebaseFirestore.instance

.collection("foods")

.doc(foodName)

.get();

if (!foodDoc.exists) {

\_show("Food not found in database!");

return;

}

final foodData = foodDoc.data()!;

final calories = int.tryParse(foodData["calories"].toString()) ?? 0;

await FirebaseFirestore.instance.collection("dailyMeals").add({

"food": foodName,

"calories": calories,

"date": today,

"createdAt": FieldValue.serverTimestamp(),

});

\_show("Meal Added!");

\_foodController.clear();

} catch (e) {

\_show("Error adding meal: $e");

}

}

void \_show(String msg) {

ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text(msg)));

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text("Add Meal"),

backgroundColor: Colors.green.shade700,

),

body: Padding(

padding: const EdgeInsets.all(20.0),

child: Column(

children: [

TextField(

controller: \_foodController,

decoration: InputDecoration(

labelText: "Food name",

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(15),

),

prefixIcon: const Icon(Icons.fastfood),

),

),

const SizedBox(height: 20),

ElevatedButton(

style: ElevatedButton.styleFrom(

backgroundColor: Colors.green,

padding:

const EdgeInsets.symmetric(horizontal: 30, vertical: 15),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(15)),

),

onPressed: \_addMeal,

child: const Text(

"Save Meal",

style: TextStyle(fontSize: 18, color: Colors.white),

),

),

],

),

),

);

}

}

class ViewMealsPage extends StatelessWidget {

const ViewMealsPage({super.key});

void \_editMeal(BuildContext context, String docId, String oldFood) {

final controller = TextEditingController(text: oldFood);

showDialog(

context: context,

builder: (ctx) {

return AlertDialog(

title: const Text("Edit Meal"),

content: TextField(

controller: controller,

decoration: const InputDecoration(labelText: "New food name"),

),

actions: [

TextButton(

onPressed: () => Navigator.pop(ctx),

child: const Text("Cancel"),

),

ElevatedButton(

onPressed: () async {

final newFood = controller.text.trim().toLowerCase();

if (newFood.isEmpty) return;

final foodDoc = await FirebaseFirestore.instance

.collection("foods")

.doc(newFood)

.get();

if (!foodDoc.exists) {

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(content: Text("Food not found!")),

);

return;

}

final foodData = foodDoc.data()!;

final calories =

int.tryParse(foodData["calories"].toString()) ?? 0;

await FirebaseFirestore.instance

.collection("dailyMeals")

.doc(docId)

.update({"food": newFood, "calories": calories});

Navigator.pop(ctx);

},

child: const Text("Save"),

),

],

);

},

);

}

void \_deleteMeal(String docId) {

FirebaseFirestore.instance.collection("dailyMeals").doc(docId).delete();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text("All Meals"),

backgroundColor: Colors.green.shade700,

),

body: StreamBuilder<QuerySnapshot>(

stream: FirebaseFirestore.instance

.collection("dailyMeals")

.orderBy("createdAt", descending: true)

.snapshots(),

builder: (context, snapshot) {

if (snapshot.hasError) {

return Center(

child: Text(

"Error: ${snapshot.error}",

style: const TextStyle(color: Colors.red),

),

);

}

if (!snapshot.hasData) {

return const Center(child: CircularProgressIndicator());

}

final docs = snapshot.data!.docs;

if (docs.isEmpty) {

return const Center(child: Text("No meals yet"));

}

return ListView.builder(

padding: const EdgeInsets.all(10),

itemCount: docs.length,

itemBuilder: (context, i) {

final doc = docs[i];

final data = doc.data() as Map<String, dynamic>;

return Card(

color: Colors.white,

elevation: 4,

margin: const EdgeInsets.symmetric(vertical: 8),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(15),

),

child: ListTile(

leading: const Icon(Icons.fastfood, color: Colors.orange),

title: Text(

"${data["food"]}",

style: const TextStyle(fontWeight: FontWeight.bold),

),

subtitle: Text("Date: ${data["date"]}"),

trailing: Row(

mainAxisSize: MainAxisSize.min,

children: [

Text(

"${data["calories"]} kcal",

style: const TextStyle(

color: Colors.blue,

fontWeight: FontWeight.bold,

),

),

IconButton(

icon: const Icon(Icons.edit, color: Colors.blue),

onPressed: () =>

\_editMeal(context, doc.id, data["food"]),

),

IconButton(

icon: const Icon(Icons.delete, color: Colors.red),

onPressed: () => \_deleteMeal(doc.id),

),

],

),

),

);

},

);

},

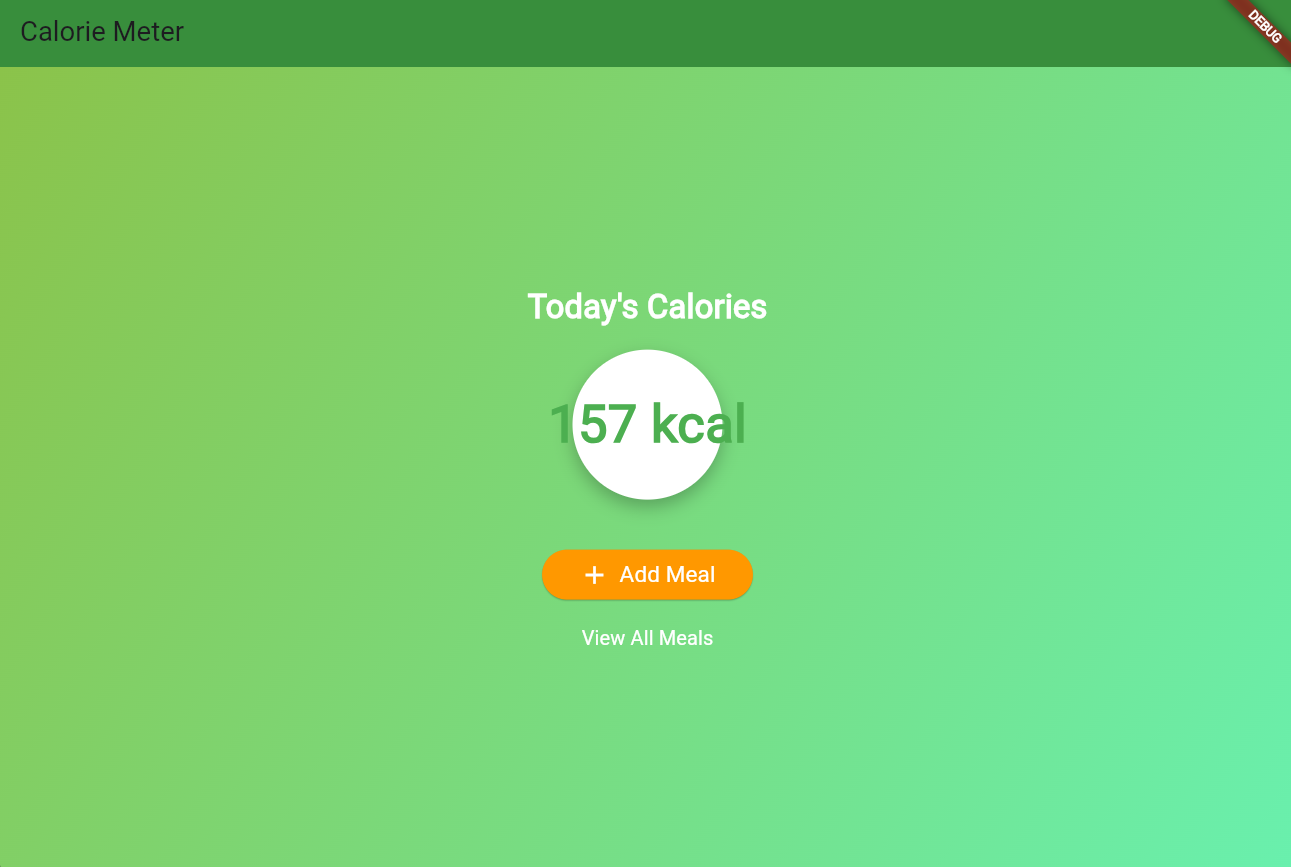
),

);

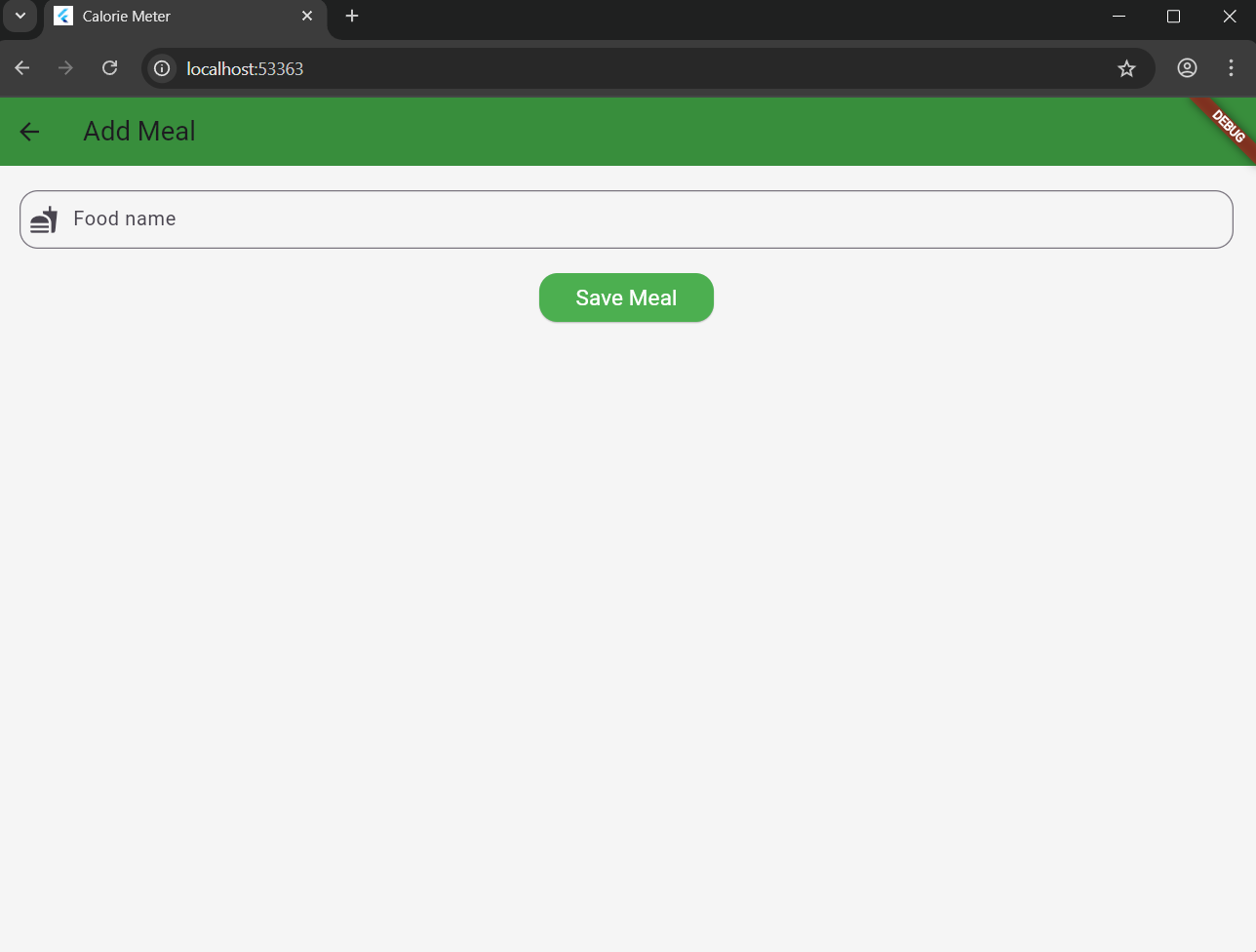
}

}

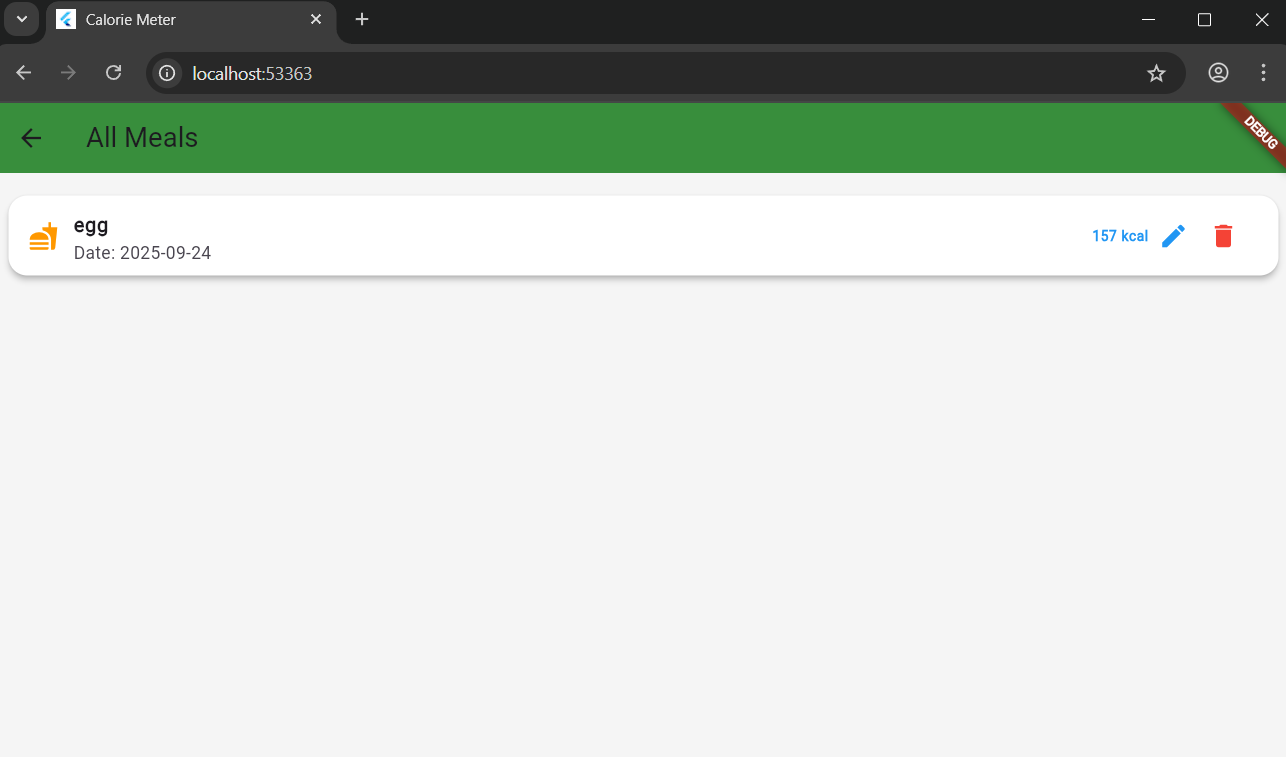
DEPENDENCIES:   
 flutter:  
 sdk: flutter  
 firebase\_core: ^4.1.0  
 cloud\_firestore: ^6.0.1  
 intl: ^0.19.0

SCREENSHOTS:

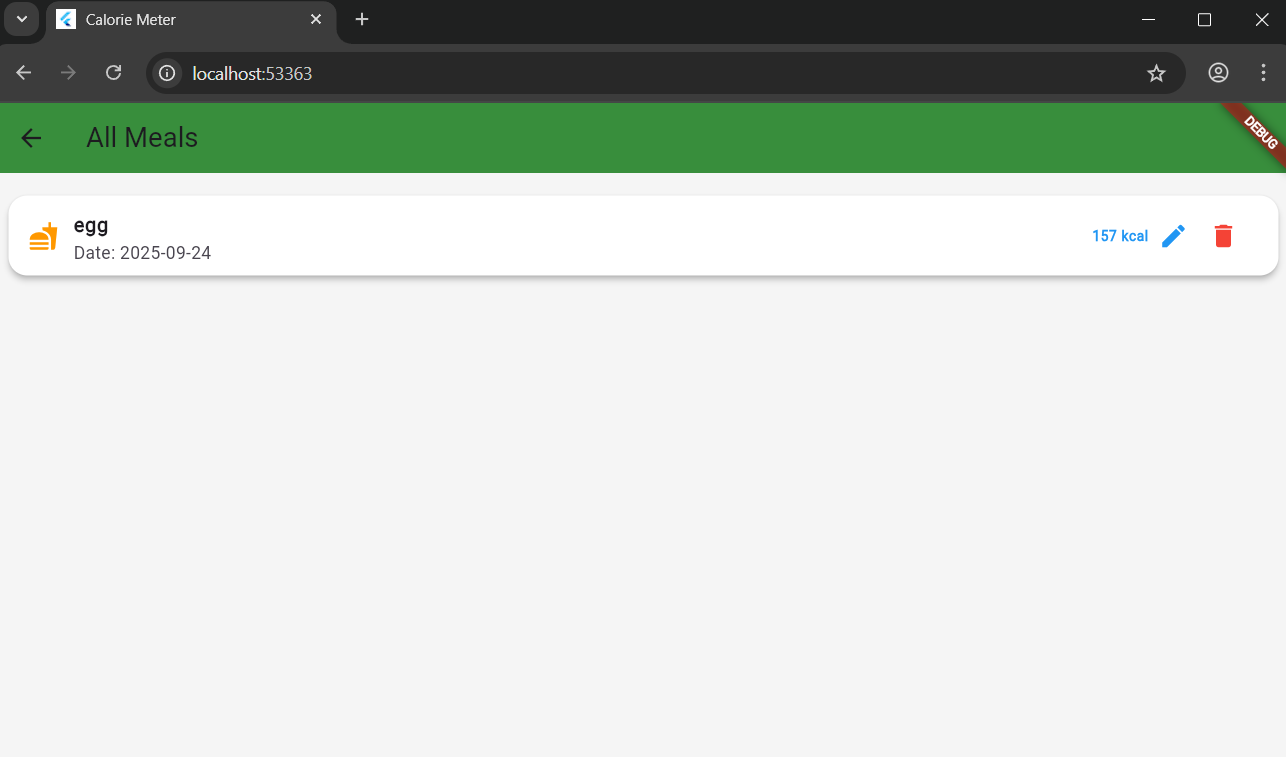
1.HOME PAGE



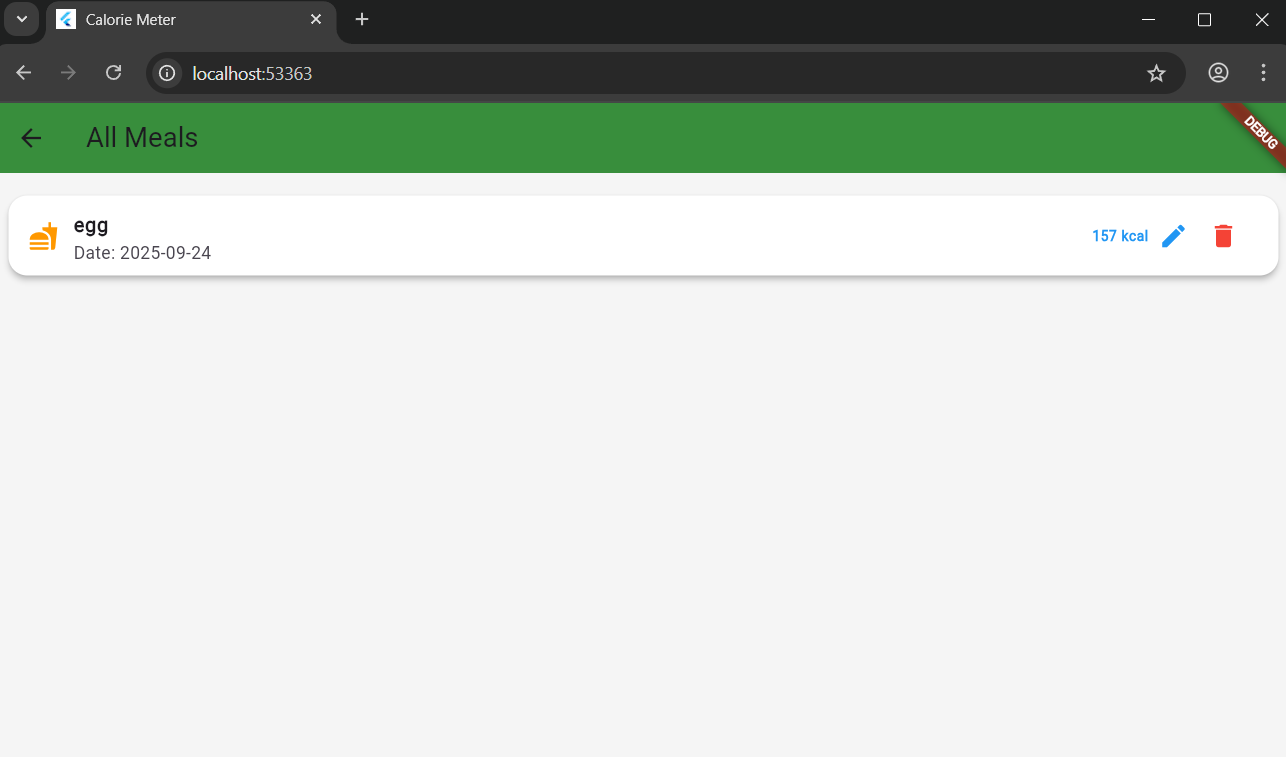
2.INSERT



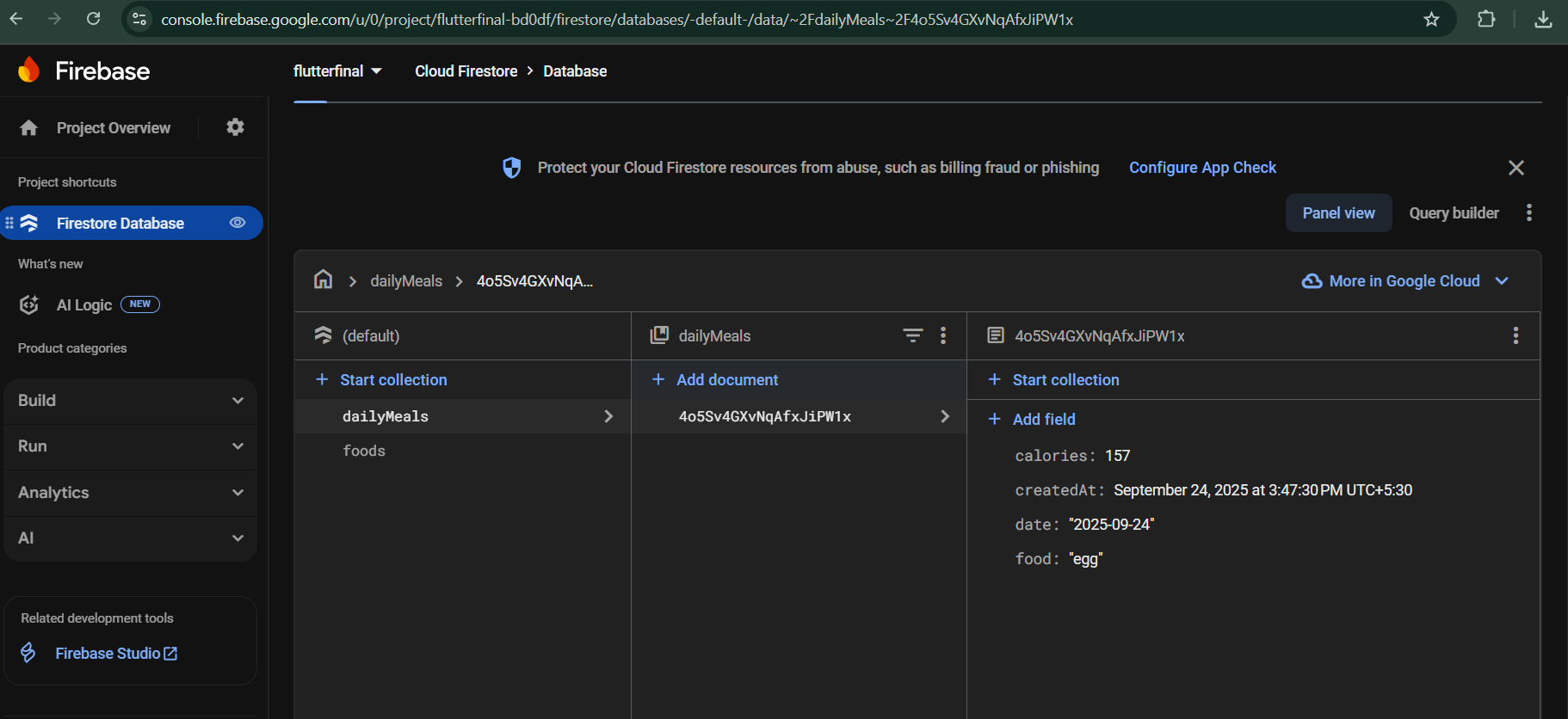
3.READ



4.UPDATE



5.DELETE



6.FIREBASE