

CODE

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

import 'fooderlich_theme.dart';
import 'models/models.dart';
import 'navigation/app_router.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  final appStateManager = AppStateManager();
  await appStateManager.initializeApp();
  runApp(Fooderlich(appStateManager: appStateManager));
}

class Fooderlich extends StatefulWidget {
  final AppStateManager appStateManager;

  const Fooderlich({
    super.key,
    required this.appStateManager,
  });

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

class FooderlichState extends State<Fooderlich> {
  late final _groceryManager = GroceryManager();
  late final _profileManager = ProfileManager();
  late final _appRouter = AppRouter(
    widget.appStateManager,
    _profileManager,
    _groceryManager,
  );

  @override
  Widget build(BuildContext context) {
    return MultiProvider(
      providers: [
        ChangeNotifierProvider(
          create: (context) => _groceryManager,
        ),
        ChangeNotifierProvider(
          create: (context) => _profileManager,
        ),
        ChangeNotifierProvider(
          create: (context) => widget.appStateManager,
        ),
      ],
      child: Consumer<ProfileManager>(
        builder: (context, profileManager, child) {
          ThemeData theme;
          if (profileManager.darkMode) {
            theme = FooderlichTheme.dark();
          } else {
            theme = FooderlichTheme.light();
          }

          final router = _appRouter.router;
```

```

        return MaterialApp.router(
          theme: theme,
          title: 'Fooderlich',
          routerDelegate: router.routerDelegate,
          routeInformationParser: router.routeInformationParser,
          routeInformationProvider: router.routeInformationProvider,
        );
      },
    ),
  );
}
}

```

```

import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';

```

```

class FooderlichTheme {
  static TextTheme lightTextTheme = TextTheme(
    bodyText1: GoogleFonts.openSans(
      fontSize: 14.0,
      fontWeight: FontWeight.w700,
      color: Colors.black,
    ),
    headline1: GoogleFonts.openSans(
      fontSize: 32.0,
      fontWeight: FontWeight.bold,
      color: Colors.black,
    ),
    headline2: GoogleFonts.openSans(
      fontSize: 21.0,
      fontWeight: FontWeight.w700,
      color: Colors.black,
    ),
    headline3: GoogleFonts.openSans(
      fontSize: 16.0,
      fontWeight: FontWeight.w600,
      color: Colors.black,
    ),
    headline6: GoogleFonts.openSans(
      fontSize: 20.0,
      fontWeight: FontWeight.w600,
      color: Colors.black,
    ),
  );
}

```

```

static TextTheme darkTextTheme = TextTheme(
  bodyText1: GoogleFonts.openSans(
    fontSize: 14.0,
    fontWeight: FontWeight.w700,
    color: Colors.white,
  ),
  headline1: GoogleFonts.openSans(
    fontSize: 32.0,
    fontWeight: FontWeight.bold,
    color: Colors.white,
  ),
  headline2: GoogleFonts.openSans(
    fontSize: 21.0,
    fontWeight: FontWeight.w700,
    color: Colors.white,
  ),
  headline3: GoogleFonts.openSans(

```

```

        fontSize: 16.0,
        fontWeight: FontWeight.w600,
        color: Colors.white,
    ),
    headline6: GoogleFonts.openSans(
        fontSize: 20.0,
        fontWeight: FontWeight.w600,
        color: Colors.white,
    ),
);

```

```

static ThemeData light() {
    return ThemeData(
        brightness: Brightness.light,
        checkboxTheme: CheckboxThemeData(
            fillColor: MaterialStateColor.resolveWith((states) {
                return Colors.black;
            }),
        ),
        appBarTheme: const AppBarTheme(
            foregroundColor: Colors.black,
            backgroundColor: Colors.white,
        ),
        floatingActionButtonTheme: const FloatingActionButtonThemeData(
            foregroundColor: Colors.white,
            backgroundColor: Colors.black,
        ),
        bottomNavigationBarTheme: const BottomNavigationBarThemeData(
            selectedItemColor: Colors.green,
        ),
        textTheme: lightTextTheme,
    );
}

```

```

static ThemeData dark() {
    return ThemeData(
        brightness: Brightness.dark,
        appBarTheme: AppBarTheme(
            foregroundColor: Colors.white,
            backgroundColor: Colors.grey[900],
        ),
        floatingActionButtonTheme: const FloatingActionButtonThemeData(
            foregroundColor: Colors.white,
            backgroundColor: Colors.green,
        ),
        bottomNavigationBarTheme: const BottomNavigationBarThemeData(
            selectedItemColor: Colors.green,
        ),
        textTheme: darkTextTheme,
    );
}
}

```

```

import 'package:flutter/material.dart';
import 'package:go_router/go_router.dart';
import '../models/models.dart';
import '../screens/screens.dart';

```

```

class AppRouter {
    final AppStateManager appStateManager;
    final ProfileManager profileManager;
    final GroceryManager groceryManager;

```

```
AppRouter(  
  this.appStateManager,  
  this.profileManager,  
  this.groceryManager,  
);
```

```
late final router = GoRouter(  
  debugLogDiagnostics: true,  
  refreshListenable: appStateManager,  
  initialLocation: '/login',  
  routes: [  
    GoRoute(  
      name: 'login',  
      path: '/login',  
      builder: (context, state) => const LoginScreen(),  
    ),  
    GoRoute(  
      name: 'onboarding',  
      path: '/onboarding',  
      builder: (context, state) => const OnboardingScreen(),  
    ),  
    GoRoute(  
      name: 'home',  
      path: '/:tab',  
      builder: (context, state) {  
        final tab = int.tryParse(state.params['tab'] ?? "") ?? 0;  
        return Home(  
          key: state.pageKey,  
          currentTab: tab,  
        );  
      },  
    ),  
    routes: [  
      GoRoute(  
        name: 'item',  
        path: 'item/:id',  
        builder: (context, state) {  
          final itemId = state.params['id'] ?? "";  
          final item = groceryManager.getGroceryItem(itemId);  
          return GroceryItemScreen(  
            originalItem: item,  
            onCreate: (item) {  
              groceryManager.addItem(item);  
            },  
            onUpdate: (item) {  
              groceryManager.updateItem(item);  
            },  
          );  
        },  
      ),  
      GoRoute(  
        name: 'profile',  
        path: 'profile',  
        builder: (context, state) {  
          final tab = int.tryParse(state.params['tab'] ?? "") ?? 0;  
          return ProfileScreen(  
            user: profileManager.getUser,  
            currentTab: tab,  
          );  
        },  
      ),  
    ],  
  ],  
  routes: [  
    GoRoute(  
      name: 'rw',  
      path: 'rw',  
      builder: (context, state) => const WebViewScreen(),
```

```

        ),
    ],
),
],
redirect: (state) {
    final loggedIn = appStateManager.isLoggedIn;
    final loggingIn = state.subloc == '/login';
    if (!loggedIn) return loggingIn ? null : '/login';

    final isOnboardingComplete = appStateManager.isOnboardingComplete;
    final onboarding = state.subloc == '/onboarding';
    if (!isOnboardingComplete) {
        return onboarding ? null : '/onboarding';
    }

    if (loggingIn || onboarding) return '${FooderlichTab.explore}';
    return null;
},
errorPageBuilder: (context, state) {
    return MaterialPage(
        key: state.pageKey,
        child: Scaffold(
            body: Center(
                child: Text(
                    state.error.toString(),
                ),
            ),
        ),
    );
},
);
}

```

```

class User {
    final String firstName;
    final String lastName;
    final String role;
    final String profileImageUrl;
    final int points;
    final bool darkMode;

    User({
        required this.firstName,
        required this.lastName,
        required this.role,
        required this.profileImageUrl,
        required this.points,
        required this.darkMode,
    });
}

```

```

class SimpleRecipe {
    String id;
    String dishImage;
    String title;
    String duration;
    String source;
    List<String> information;

    SimpleRecipe({
        required this.id,
        required this.dishImage,
    }) {}
}

```

```

        required this.title,
        required this.duration,
        required this.source,
        required this.information,
    });

factory SimpleRecipe.fromJson(Map<String, dynamic> json) {
  return SimpleRecipe(
    id: json['id'] as String,
    dishImage: json['dishImage'] as String,
    title: json['title'] as String,
    duration: json['duration'] as String,
    source: json['source'] as String,
    information: json['information'].cast<String>() as List<String>,
  );
}

import 'package:flutter/material.dart';

import 'models.dart';

class ProfileManager extends ChangeNotifier {
  User get getUser => User(
    firstName: 'Missy Mae Ferrer Sanchez',
    lastName: 'Patt',
    role: 'IT Manager',
    profileImageUrl: 'assets/profile_pics/person_cesare.jpg',
    points: 100,
    darkMode: _darkMode,
  );

  bool get didSelectUser => _didSelectUser;
  bool get didTapOnRaywenderlich => _tapOnRaywenderlich;
  bool get darkMode => _darkMode;

  var _didSelectUser = false;
  var _tapOnRaywenderlich = false;
  var _darkMode = false;

  set darkMode(bool darkMode) {
    _darkMode = darkMode;
    notifyListeners();
  }

  void tapOnRaywenderlich(bool selected) {
    _tapOnRaywenderlich = selected;
    notifyListeners();
  }

  void tapOnProfile(bool selected) {
    _didSelectUser = selected;
    notifyListeners();
  }
}

class Post {
  String id;
  String profileImageUrl;
  String comment;
  String foodPictureUrl;
  String timestamp;

```

```

Post({
  required this.id,
  required this.imageUrl,
  required this.comment,
  required this.foodPictureUrl,
  required this.timestamp,
});

factory Post.fromJson(Map<String, dynamic> json) {
  return Post(
    id: json['id'] ?? "",
    imageUrl: json['imageUrl'] ?? "",
    comment: json['comment'] ?? "",
    foodPictureUrl: json['foodPictureUrl'] ?? "",
    timestamp: json['timestamp'] ?? "",
  );
}

export 'explore_data.dart';
export 'explore_recipe.dart';
export 'grocery_item.dart';
export 'grocery_manager.dart';
export 'post.dart';
export 'profile_manager.dart';
export 'simple_recipe.dart';
export 'user.dart';
export 'app_cache.dart';
export 'app_state_manager.dart';

part of 'explore_recipe.dart';

class Instruction {
  String imageUrl;
  String description;
  int durationInMinutes;

  Instruction({
    required this.imageUrl,
    required this.description,
    required this.durationInMinutes,
  });

  factory Instruction.fromJson(Map<String, dynamic> json) {
    return Instruction(
      imageUrl: json['imageUrl'] ?? "",
      description: json['description'] ?? "",
      durationInMinutes: json['durationInMinutes'] ?? "",
    );
  }
}

part of 'explore_recipe.dart';

class Ingredients {
  String imageUrl;
  String title;
  String source;

  Ingredients({
    required this.imageUrl,
    required this.title,
    required this.source,
  });
}

```

```

});

factory Ingredients.fromJson(Map<String, dynamic> json) {
  return Ingredients(
    imageUrl: json['imageUrl'] ?? "",
    title: json['title'] ?? "",
    source: json['source'] ?? "",
  );
}

import 'package:flutter/material.dart';

import 'grocery_item.dart';

class GroceryManager extends ChangeNotifier {
  final _groceryItems = <GroceryItem>[];
  List<GroceryItem> get groceryItems => List.unmodifiable(_groceryItems);

  void deleteItem(int index) {
    _groceryItems.removeAt(index);
    notifyListeners();
  }

  String getItemId(int index) {
    final groceryItem = _groceryItems[index];
    return groceryItem.id;
  }

  GroceryItem? getGroceryItem(String id) {
    final index = _groceryItems.indexWhere((element) => element.id == id);
    if (index == -1) return null;
    return groceryItems[index];
  }

  void addItem(GroceryItem item) {
    _groceryItems.add(item);
    notifyListeners();
  }

  void updateItem(GroceryItem item) {
    final index = _groceryItems.indexWhere((element) => element.id == item.id);
    _groceryItems[index] = item;
    notifyListeners();
  }

  void completeItem(int index, bool change) {
    final item = _groceryItems[index];
    _groceryItems[index] = item.copyWith(isComplete: change);
    notifyListeners();
  }
}

import 'package:flutter/painting.dart';

enum Importance { low, medium, high }

class GroceryItem {
  final String id;
  final String name;
  final Importance importance;
  final Color color;
  final int quantity;

```



```
final DateTime date;
final bool isComplete;
```

```
GroceryItem({
  required this.id,
  required this.name,
  required this.importance,
  required this.color,
  required this.quantity,
  required this.date,
  this.isComplete = false,
});
```

```
GroceryItem copyWith(
  {String? id,
  String? name,
  Importance? importance,
  Color? color,
  int? quantity,
  DateTime? date,
  bool? isComplete}) {
return GroceryItem(
  id: id ?? this.id,
  name: name ?? this.name,
  importance: importance ?? this.importance,
  color: color ?? this.color,
  quantity: quantity ?? this.quantity,
  date: date ?? this.date,
  isComplete: isComplete ?? this.isComplete,
);
}
```

```
part 'ingredient.dart';
part 'instruction.dart';
```

```
class RecipeCardType {
  static const card1 = 'card1';
  static const card2 = 'card2';
  static const card3 = 'card3';
}
```

```
class ExploreRecipe {
  String id;
  String cardType;
  String title;
  String subtitle;
  String backgroundImage;
  String backgroundImageSource;
  String message;
  String authorName;
  String role;
  String profileImage;
  int durationInMinutes;
  String dietType;
  int calories;
  List<String> tags;
  String description;
  String source;
  List<Ingredients> ingredients;
  List<Instruction> instructions;
```

```
ExploreRecipe({
```

```
required this.id,  
required this.cardType,  
required this.title,  
this.subtitle = "",  
this.backgroundImage = "",  
this.backgroundImageSource = "",  
this.message = "",  
this.authorName = "",  
this.role = "",  
this.profileImage = "",  
this.durationInMinutes = 0,  
this.dietType = "",  
this.calories = 0,  
this.tags = const [],  
this.description = "",  
this.source = "",  
this.ingredients = const [],  
this.instructions = const [],  
});
```

```
factory ExploreRecipe.fromJson(Map<String, dynamic> json) {  
  final ingredients = <Ingredients>[];  
  final instructions = <Instruction>[];
```

```
  if (json['ingredients'] != null) {  
    json['ingredients'].forEach((v) {  
      ingredients.add(Ingredients.fromJson(v));  
    });  
  }
```

```
  if (json['instructions'] != null) {  
    json['instructions'].forEach((v) {  
      instructions.add(Instruction.fromJson(v));  
    });  
  }
```

```
  return ExploreRecipe(  
    id: json['id'] ?? "",  
    cardType: json['cardType'] ?? "",  
    title: json['title'] ?? "",  
    subtitle: json['subtitle'] ?? "",  
    backgroundImage: json['backgroundImage'] ?? "",  
    backgroundImageSource: json['backgroundImageSource'] ?? "",  
    message: json['message'] ?? "",  
    authorName: json['authorName'] ?? "",  
    role: json['role'] ?? "",  
    profileImage: json['profileImage'] ?? "",  
    durationInMinutes: json['durationInMinutes'] ?? 0,  
    dietType: json['dietType'] ?? "",  
    calories: json['calories'] ?? 0,  
    tags: json['tags'].cast<String>() ?? [],  
    description: json['description'] ?? "",  
    source: json['source'] ?? "",  
    ingredients: ingredients,  
    instructions: instructions,  
  );  
}
```

```
import 'dart:async';  
import 'package:flutter/material.dart';  
import 'models.dart';
```

```

class FooderlichTab {
    static const int explore = 0;
    static const int recipes = 1;
    static const int toBuy = 2;
}

// AppStateManager mocks the various app state such as app initialization,
// app login, and onboarding.
class AppStateManager extends ChangeNotifier {
    // Checks to see if the user is logged in
    bool _loggedIn = false;
    // Checks to see if the user has completed onboarding
    bool _onboardingComplete = false;
    // Records the current tab the user is on.
    int _selectedTab = FooderlichTab.explore;
    // Stores user state properties on platform specific file system.
    final _appCache = AppCache();

    // Property getters.
    bool get isLoggedIn => _loggedIn;
    bool get isOnboardingComplete => _onboardingComplete;
    int get getSelectedTab => _selectedTab;

    // Initializes the app
    Future<void> initializeApp() async {
        // Check if the user is logged in
        _loggedIn = await _appCache.isUserLoggedIn();
        // Check if the user completed onboarding
        _onboardingComplete = await _appCache.didCompleteOnboarding();
    }

    void login(String username, String password) async {
        _loggedIn = true;
        await _appCache.cacheUser();
        notifyListeners();
    }

    void onboarded() async {
        _onboardingComplete = true;
        await _appCache.completeOnboarding();
        notifyListeners();
    }

    void goToTab(index) {
        _selectedTab = index;
        notifyListeners();
    }

    void goToRecipes() {
        _selectedTab = FooderlichTab.recipes;
        notifyListeners();
    }

    void logout() async {
        // Reset all properties once user logs out
        _loggedIn = false;
        _onboardingComplete = false;
        _selectedTab = 0;

        // Reinitialize the app
        await _appCache.invalidate();
        await initializeApp();
        notifyListeners();
    }

```

```
}  
}
```

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

```
class AppCache {  
  static const kUser = 'user';  
  static const kOnboarding = 'onboarding';  
  
  Future<void> invalidate() async {  
    final prefs = await SharedPreferences.getInstance();  
    await prefs.setBool(kUser, false);  
    await prefs.setBool(kOnboarding, false);  
  }  
  
  Future<void> cacheUser() async {  
    final prefs = await SharedPreferences.getInstance();  
    await prefs.setBool(kUser, true);  
  }  
  
  Future<void> completeOnboarding() async {  
    final prefs = await SharedPreferences.getInstance();  
    await prefs.setBool(kOnboarding, true);  
  }  
  
  Future<bool> isUserLoggedIn() async {  
    final prefs = await SharedPreferences.getInstance();  
    return prefs.getBool(kUser) ?? false;  
  }  
  
  Future<bool> didCompleteOnboarding() async {  
    final prefs = await SharedPreferences.getInstance();  
    return prefs.getBool(kOnboarding) ?? false;  
  }  
}
```

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

```
import '../components/components.dart';  
import '../models/models.dart';
```

```
class TodayRecipeListView extends StatelessWidget {  
  final List<ExploreRecipe> recipes;  
  
  const TodayRecipeListView({  
    super.key,  
    required this.recipes,  
  });  
  
  @override  
  Widget build(BuildContext context) {  
    return Padding(  
      padding: const EdgeInsets.only(  
        left: 16,  
        right: 16,  
        top: 16,  
      ),  
      child: Column(  
        crossAxisAlignment: CrossAxisAlignment.start,  
        children: [  
          Text(  
            'Recipes of the Day 🍷',  
            style: Theme.of(context).textTheme.headline1,
```

```

    ),
    const SizedBox(height: 16),
    Container(
      height: 400,
      color: Colors.transparent,
      child: ListView.separated(
        scrollDirection: Axis.horizontal,
        itemCount: recipes.length,
        itemBuilder: (context, index) {
          final recipe = recipes[index];
          return buildCard(recipe);
        },
        separatorBuilder: (context, index) {
          return const SizedBox(width: 16);
        },
      ),
    ),
  ],
),
);
}

```

```

Widget buildCard(ExploreRecipe recipe) {
  if (recipe.cardType == RecipeCardType.card1) {
    return Card1(recipe: recipe);
  } else if (recipe.cardType == RecipeCardType.card2) {
    return Card2(recipe: recipe);
  } else if (recipe.cardType == RecipeCardType.card3) {
    return Card3(recipe: recipe);
  } else {
    throw Exception("This card doesn't exist yet");
  }
}
}

```

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

```
import '../components/components.dart';
```

```
import '../models/models.dart';
```

```
class RecipesGridView extends StatelessWidget {
  final List<SimpleRecipe> recipes;
```

```

  const RecipesGridView({
    super.key,
    required this.recipes,
  });

```

```
@override
```

```

Widget build(BuildContext context) {
  return Padding(
    padding: const EdgeInsets.only(
      left: 16,
      right: 16,
      top: 16,
    ),
    child: GridView.builder(
      itemCount: recipes.length,
      gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: 2,
      ),
      itemBuilder: (context, index) {
        final simpleRecipe = recipes[index];

```

```

        return RecipeThumbnail(recipe: simpleRecipe);
      },
    ),
  );
}
}

```

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

```
import '../models/models.dart';
```

```
class RecipeThumbnail extends StatelessWidget {
  final SimpleRecipe recipe;
```

```

  const RecipeThumbnail({
    super.key,
    required this.recipe,
  });

```

```
@override
```

```

Widget build(BuildContext context) {
  return Container(
    padding: const EdgeInsets.all(8),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Expanded(
          child: ClipRRect(
            borderRadius: BorderRadius.circular(12),
            child: Image.asset(
              recipe.dishImage,
              fit: BoxFit.cover,
            ),
          ),
        ),
        const SizedBox(height: 10),
        Text(
          recipe.title,
          maxLines: 1,
          style: Theme.of(context).textTheme.bodyText1,
        ),
        Text(
          recipe.duration,
          style: Theme.of(context).textTheme.bodyText1,
        ),
      ],
    ),
  );
}
}

```

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

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

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

```
import '../models/grocery_item.dart';
```

```

class GroceryTile extends StatelessWidget {
  final GroceryItem item;
  final Function(bool?)? onComplete;
  final TextDecoration textDecoration;

```

```

  GroceryTile({

```

```

super.key,
required this.item,
this.onComplete,
}) : textDecoration =
    item.isComplete ? TextDecoration.lineThrough : TextDecoration.none;

```

```

@override

```

```

Widget build(BuildContext context) {
  return SizedBox(
    height: 100,
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Row(
          children: [
            Container(
              width: 5.0,
              color: item.color,
            ),
            const SizedBox(width: 16.0),
            Column(
              mainAxisAlignment: MainAxisAlignment.center,
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  item.name,
                  style: GoogleFonts.lato(
                    decoration: textDecoration,
                    fontSize: 21.0,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                const SizedBox(height: 4.0),
                buildDate(),
                const SizedBox(height: 4.0),
                buildImportance(),
              ],
            ),
          ],
        ),
        Row(
          children: [
            Text(
              item.quantity.toString(),
              style: GoogleFonts.lato(
                decoration: textDecoration,
                fontSize: 21,
              ),
            ),
            buildCheckbox()
          ],
        ),
      ],
    );
}

```

```

Widget buildImportance() {
  if (item.importance == Importance.low) {
    return Text(
      'Low',
      style: GoogleFonts.lato(decoration: textDecoration),
    );
  }
}

```

```

} else if (item.importance == Importance.medium) {
  return Text(
    'Medium',
    style: GoogleFonts.lato(
      fontWeight: FontWeight.w800, decoration: textDecoration),
  );
} else if (item.importance == Importance.high) {
  return Text(
    'High',
    style: GoogleFonts.lato(
      color: Colors.red,
      fontWeight: FontWeight.w900,
      decoration: textDecoration,
    ),
  );
} else {
  throw Exception("This importance type does not exist");
}
}

```

```

Widget buildDate() {
  final dateFormatter = DateFormat('MMMM dd h:mm a');
  final dateString = dateFormatter.format(item.date);
  return Text(
    dateString,
    style: TextStyle(decoration: textDecoration),
  );
}

```

```

Widget buildCheckbox() {
  return Checkbox(
    value: item.isComplete,
    onChanged: onComplete,
  );
}
}

```

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

```
import '../components/components.dart';
import '../models/models.dart';
```

```
class FriendPostTile extends StatelessWidget {
  final Post post;
```

```

  const FriendPostTile({
    super.key,
    required this.post,
  });

```

```
@override
```

```

Widget build(BuildContext context) {
  return Row(
    crossAxisAlignment: CrossAxisAlignment.start,
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
      CircleImage(
        imageProvider: AssetImage(post.profileImageUrl),
        imageRadius: 20,
      ),
      const SizedBox(width: 16),
      Expanded(
        child: Column(

```



```

        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(post.comment),
          Text(
            '${post.timestamp} mins ago',
            style: Theme.of(context).textTheme.bodyText1,
          ),
        ],
      ),
    ),
  ],
);
}
}

```

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

```
import '../components/components.dart';
import '../models/models.dart';
```

```
class FriendPostListView extends StatelessWidget {
  final List<Post> friendPosts;
```

```

  const FriendPostListView({
    super.key,
    required this.friendPosts,
  });

```

```
@override
```

```

Widget build(BuildContext context) {
  return Padding(
    padding: const EdgeInsets.only(
      left: 16,
      right: 16,
      top: 0,
    ),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(
          'Social Chefs • ',
          style: Theme.of(context).textTheme.headline1,
        ),
        const SizedBox(height: 16),
        ListView.separated(
          primary: false,
          physics: const NeverScrollableScrollPhysics(),
          shrinkWrap: true,
          scrollDirection: Axis.vertical,
          itemCount: friendPosts.length,
          itemBuilder: (context, index) {
            final post = friendPosts[index];
            return FriendPostTile(post: post);
          },
          separatorBuilder: (context, index) {
            return const SizedBox(height: 16);
          },
        ),
        const SizedBox(height: 16),
      ],
    ),
  );
}

```

```
}
```

```
export 'author_card.dart';
export 'card1.dart';
export 'card2.dart';
export 'card3.dart';
export 'circle_image.dart';
export 'friend_post_list_view.dart';
export 'friend_post_tile.dart';
export 'recipe_thumbnail.dart';
export 'recipes_grid_view.dart';
export 'today_recipe_list_view.dart';
```

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

```
class CircleImage extends StatelessWidget {
  const CircleImage({
    super.key,
    this.imageProvider,
    this.imageRadius = 20,
  });
```

```
  final double imageRadius;
  final ImageProvider? imageProvider;
```

```
  @override
```

```
  Widget build(BuildContext context) {
    return CircleAvatar(
      backgroundColor: Colors.white,
      radius: imageRadius,
      child: CircleAvatar(
        radius: imageRadius - 5,
        backgroundImage: imageProvider,
      ),
    );
  }
}
```

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

```
import '../fooderlich_theme.dart';
import '../models/models.dart';
```

```
class Card3 extends StatelessWidget {
  final ExploreRecipe recipe;
```

```
  const Card3({
    super.key,
    required this.recipe,
  });
```

```
  List<Widget> createTagChips() {
    final chips = <Widget>[];
    recipe.tags.take(6).forEach(
      (element) {
        final chip = Chip(
          label: Text(
            element,
            style: FooderlichTheme.darkTextTheme.bodyText1,
          ),
          backgroundColor: Colors.black.withOpacity(0.7),
        );
        chips.add(chip);
      }
    );
  }
}
```

```
    },
  );

  return chips;
}
```

```
@override
Widget build(BuildContext context) {
  return Center(
    child: Container(
      constraints: const BoxConstraints.expand(
        width: 350,
        height: 450,
      ),
      decoration: BoxDecoration(
        image: DecorationImage(
          image: AssetImage(recipe.backgroundImage),
          fit: BoxFit.cover,
        ),
        borderRadius: const BorderRadius.all(
          Radius.circular(10.0),
        ),
      ),
    ),
    child: Stack(
      children: [
        Container(
          decoration: BoxDecoration(
            color: Colors.black.withOpacity(0.6),
            borderRadius: const BorderRadius.all(
              Radius.circular(10.0),
            ),
          ),
        ),
        Container(
          padding: const EdgeInsets.all(16),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              const Icon(
                Icons.book,
                color: Colors.white,
                size: 40,
              ),
              const SizedBox(height: 8),
              Text(
                recipe.title,
                style: FooderlichTheme.darkTextTheme.headline2,
              ),
              const SizedBox(height: 30),
            ],
          ),
        ),
      ],
    ),
  ),
  Center(
    child: Wrap(
      alignment: WrapAlignment.start,
      spacing: 12,
      runSpacing: 12,
      children: createTagChips(),
    ),
  ),
),
```

```

    );
  }
}
import 'package:flutter/material.dart';

import '../fooderlich_theme.dart';
import '../models/models.dart';
import 'author_card.dart';

class Card2 extends StatelessWidget {
  final ExploreRecipe recipe;

  const Card2({
    super.key,
    required this.recipe,
  });

  @override
  Widget build(BuildContext context) {
    return Center(
      child: Container(
        constraints: const BoxConstraints.expand(
          width: 350,
          height: 450,
        ),
        decoration: BoxDecoration(
          image: DecorationImage(
            image: AssetImage(recipe.backgroundImage),
            fit: BoxFit.cover,
          ),
          borderRadius: const BorderRadius.all(
            Radius.circular(10.0),
          ),
        ),
      child: Column(
        children: [
          AuthorCard(
            authorName: recipe.authorName,
            title: recipe.role,
            imageProvider: AssetImage(recipe.profileImage),
          ),
          Expanded(
            child: Stack(
              children: [
                Positioned(
                  bottom: 16,
                  right: 16,
                  child: Text(
                    recipe.title,
                    style: FooderlichTheme.lightTextTheme.headline1,
                  ),
                ),
                Positioned(
                  bottom: 70,
                  left: 16,
                  child: RotatedBox(
                    quarterTurns: 3,
                    child: Text(
                      recipe.subtitle,
                      style: FooderlichTheme.lightTextTheme.headline1,
                    ),
                  ),
                ),
              ],
            ),
          ),
        ],
      ),
    );
  }
}

```

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

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

```
import '../fooderlich_theme.dart';
```

```
import '../models/models.dart';
```

```
class Card1 extends StatelessWidget {  
  final ExploreRecipe recipe;
```

```
  
  const Card1({  
    super.key,  
    required this.recipe,  
  });
```

```
  @override
```

```
  Widget build(BuildContext context) {  
    return Center(  
      child: Container(  
        padding: const EdgeInsets.all(16),  
        constraints: const BoxConstraints.expand(  
          width: 350,  
          height: 450,  
        ),  
        decoration: BoxDecoration(  
          image: DecorationImage(  
            image: AssetImage(recipe.backgroundImage),  
            fit: BoxFit.cover,  
          ),  
          borderRadius: const BorderRadius.all(  
            Radius.circular(10.0),  
          ),  
        ),  
      child: Stack(  
        children: [  
          Text(  
            recipe.subtitle,  
            style: FooderlichTheme.darkTextTheme.bodyText1,  
          ),  
          Positioned(  
            top: 20,  
            child: Text(  
              recipe.title,  
              style: FooderlichTheme.darkTextTheme.headline2,  
            ),  
          ),  
          Positioned(  
            bottom: 30,  
            right: 0,  
            child: Text(  
              recipe.message,  
              style: FooderlichTheme.darkTextTheme.bodyText1,  
            ),  
          ),  
          Positioned(  

```

```

        bottom: 10,
        right: 0,
        child: Text(
          recipe.authorName,
          style: FooderlichTheme.darkTextTheme.bodyText1,
        ),
      ),
    ],
  ),
);
}
}

```

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

```
import '../fooderlich_theme.dart';
```

```
import 'circle_image.dart';
```

```

class AuthorCard extends StatefulWidget {
  final String authorName;
  final String title;
  final ImageProvider? imageProvider;

```

```

  const AuthorCard({
    super.key,
    required this.authorName,
    required this.title,
    this.imageProvider,
  });

```

```

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

```

```

class AuthorCardState extends State<AuthorCard> {
  bool _isFavorited = false;

```

```

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: const EdgeInsets.all(16),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Row(children: [
            CircleImage(
              imageProvider: widget.imageProvider,
              imageRadius: 28,
            ),
            const SizedBox(width: 8),
            Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  widget.authorName,
                  style: FooderlichTheme.lightTextTheme.headline2,
                ),
                Text(
                  widget.title,
                  style: FooderlichTheme.lightTextTheme.headline3,
                )
              ],
            ),
          ],
        ),
      ),
    );
  }
}

```

```

    ),
  ],
  IconButton(
    icon: Icon(_isFavorited ? Icons.favorite : Icons.favorite_border),
    iconSize: 30,
    color: Colors.red[400],
    onPressed: () {
      setState(() {
        _isFavorited = !_isFavorited;
      });
    },
  ),
),
],
),
);
}
}

```

```

[{"family": "MaterialIcons", "fonts": [{"asset": "fonts/MaterialIcons-Regular.otf"}]}]
{"assets/food_pics/food_banana.jpg":["assets/food_pics/food_banana.jpg"], "assets/food_pics/food_brussels_sprouts.jpg":["assets/food_pics/food_brussels_sprouts.jpg"], "assets/food_pics/food_burger.jpg":["assets/food_pics/food_burger.jpg"], "assets/food_pics/food_cucumber.jpg":["assets/food_pics/food_cucumber.jpg"], "assets/food_pics/food_cupcake.jpg":["assets/food_pics/food_cupcake.jpg"], "assets/food_pics/food_curry.jpg":["assets/food_pics/food_curry.jpg"], "assets/food_pics/food_dozen_eggs.jpg":["assets/food_pics/food_dozen_eggs.jpg"], "assets/food_pics/food_flour.jpg":["assets/food_pics/food_flour.jpg"], "assets/food_pics/food_friedrice.jpg":["assets/food_pics/food_friedrice.jpg"], "assets/food_pics/food_green_beans.jpg":["assets/food_pics/food_green_beans.jpg"], "assets/food_pics/food_pancake.jpg":["assets/food_pics/food_pancake.jpg"], "assets/food_pics/food_peanutbutter.png":["assets/food_pics/food_peanutbutter.png"], "assets/food_pics/food_pho.jpg":["assets/food_pics/food_pho.jpg"], "assets/food_pics/food_salmon.jpg":["assets/food_pics/food_salmon.jpg"], "assets/food_pics/food_soymilk.png":["assets/food_pics/food_soymilk.png"], "assets/food_pics/food_spaghetti.jpg":["assets/food_pics/food_spaghetti.jpg"], "assets/fooderlich_assets/empty_list.png":["assets/fooderlich_assets/empty_list.png"], "assets/fooderlich_assets/list.png":["assets/fooderlich_assets/list.png"], "assets/fooderlich_assets/recommend.png":["assets/fooderlich_assets/recommend.png"], "assets/fooderlich_assets/rw_logo.png":["assets/fooderlich_assets/rw_logo.png"], "assets/fooderlich_assets/sheet.png":["assets/fooderlich_assets/sheet.png"], "assets/magazine_pics/card_bread.jpg":["assets/magazine_pics/card_bread.jpg"], "assets/magazine_pics/card_carrot.png":["assets/magazine_pics/card_carrot.png"], "assets/magazine_pics/card_salad.png":["assets/magazine_pics/card_salad.png"], "assets/magazine_pics/card_smoothie.png":["assets/magazine_pics/card_smoothie.png"], "assets/magazine_pics/mag1.png":["assets/magazine_pics/mag1.png"], "assets/magazine_pics/mag2.png":["assets/magazine_pics/mag2.png"], "assets/magazine_pics/mag3.png":["assets/magazine_pics/mag3.png"], "assets/magazine_pics/mag4.png":["assets/magazine_pics/mag4.png"], "assets/magazine_pics/mag5.png":["assets/magazine_pics/mag5.png"], "assets/profile_pics/person_cesare.jpg":["assets/profile_pics/person_cesare.jpg"], "assets/profile_pics/person_crispy.jpg":["assets/profile_pics/person_crispy.jpg"], "assets/profile_pics/person_ian.jpg":["assets/profile_pics/person_ian.jpg"], "assets/profile_pics/person_joe.jpg":["assets/profile_pics/person_joe.jpg"], "assets/profile_pics/person_katz.jpg":["assets/profile_pics/person_katz.jpg"], "assets/profile_pics/person_kevin.jpg":["assets/profile_pics/person_kevin.jpg"], "assets/profile_pics/person_manda.jpg":["assets/profile_pics/person_manda.jpg"], "assets/profile_pics/person_ray.jpg":["assets/profile_pics/person_ray.jpg"], "assets/profile_pics/person_sandra.jpg":["assets/profile_pics/person_sandra.jpg"], "assets/profile_pics/person_stef.jpeg":["assets/profile_pics/person_cesare.jpg"], "assets/profile_pics/person_tiffani.jpg":["assets/profile_pics/person_tiffani.jpg"], "assets/sample_data/sample_explore_recipes.json":["assets/sample_data/sample_explore_recipes.json"], "assets/sample_data/sample_friends_feed.json":["assets/sample_data/sample_friends_feed.json"], "assets/sample_data/sample_recipes.json":["assets/sample_data/sample_recipes.json"]}

```

```

{
  "recipes": [
    {
      "id": "rec1",
      "dishImage": "assets/food_pics/food_salmon.jpg",
      "title": "Smoked Salmon",
      "duration": "30 mins",
      "source": "https://pixabay.com/photos/salmon-dish-food-meal-fish-518032/",
      "information": [
        "Course: Main Course",
        "Price: Cheap",
        "Cuisine: Meditarrian",

```

```
    "Skill Level: Beginner",
    "Prep Time 5 minutes"
  ]
},
{
  "id": "rec2",
  "dishImage": "assets/food_pics/food_cupcake.jpg",
  "title": "Chocolate Cup Cake",
  "duration": "1 hour",
  "source": "https://pixabay.com/photos/cupcakes-dessert-frosting-food-690040/",
  "information": [
    "Course: Dessert",
    "Price: Somewhat cheap",
    "Cuisine: American",
    "Skill Level: Amateur",
    "Prep Time 10 minutes"
  ]
},
{
  "id": "rec3",
  "dishImage": "assets/food_pics/food_pancake.jpg",
  "title": "Rasberry Maple Pancake",
  "duration": "30 mins",
  "source": "https://pixabay.com/photos/pancakes-maple-syrup-sweet-food-2291908/",
  "information": [
    "Course: Breakfast",
    "Price: Cheap",
    "Cuisine: American",
    "Skill Level: Beginner",
    "Prep Time 5 minutes"
  ]
},
{
  "id": "rec4",
  "dishImage": "assets/food_pics/food_spaghetti.jpg",
  "title": "Seafood Spaghetti",
  "duration": "50 mins",
  "source": "https://pixabay.com/photos/spaghetti-drunken-line-food-pasta-960978/",
  "information": [
    "Course: Main Course",
    "Price: Expensive",
    "Cuisine: Italian",
    "Skill Level: Beginner",
    "Prep Time 5 minutes"
  ]
},
{
  "id": "rec5",
  "dishImage": "assets/food_pics/food_friedrice.jpg",
  "title": "Thai Fried Rice",
  "duration": "35 mins",
  "source": "https://pixabay.com/photos/fried-rice-thai-food-thai-cuisine-3023040/",
  "information": [
    "Course: Main Course",
    "Price: Cheap",
    "Cuisine: Thai",
    "Skill Level: Beginner",
    "Prep Time 5 minutes"
  ]
},
{
  "id": "rec6",
```



```
"dishImage": "assets/food_pics/food_curry.jpg",
"title": "Thai Green Curry",
"duration": "1 hour",
"source": "https://pixabay.com/photos/green-curry-thai-food-3604721/",
"information": [
  "Course: Main Course",
  "Price: Medium",
  "Cuisine: Thai",
  "Skill Level: Amateur",
  "Prep Time 10 minutes"
]
},
{
  "id": "rec7",
  "dishImage": "assets/food_pics/food_pho.jpg",
  "title": "Vietnamese Pho",
  "duration": "1 hour",
  "source": "https://pixabay.com/photos/soup-pho-soup-vietnam-vietnamese-5250765/",
  "information": [
    "Course: Main Course",
    "Price: Cheap",
    "Cuisine: Vietnamese",
    "Skill Level: Hard",
    "Prep Time 20 minutes"
  ]
},
{
  "id": "rec8",
  "dishImage": "assets/food_pics/food_burger.jpg",
  "title": "Bacon Cheese Burger",
  "duration": "30 min",
  "source": "https://pixabay.com/photos/burger-fast-food-hamburger-731298/",
  "information": [
    "Course: Main Course",
    "Price: Cheap",
    "Cuisine: American",
    "Skill Level: Easy",
    "Prep Time 5 minutes"
  ]
}
]
}
{
  "feed": [
    {
      "id": "1",
      "profileImageUrl": "assets/profile_pics/person_cesare.jpg",
      "comment": "Made this delicious pizza this morning",
      "foodPictureUrl": "card_carrot.png",
      "timestamp": "10"
    },
    {
      "id": "2",
      "profileImageUrl": "assets/profile_pics/person_ian.jpg",
      "comment": "Made delicious creamy, shrimp black pepper pasta.",
      "foodPictureUrl": "card_carrot.png",
      "timestamp": "80"
    },
    {
      "id": "3",
      "profileImageUrl": "assets/profile_pics/person_crispy.jpg",
      "comment": "Made a delicious salad today!",
      "foodPictureUrl": "card_carrot.png",
    }
  ]
}
```

```

    "timestamp": "20"
  },
  {
    "id": "4",
    "profileImageUrl": "assets/profile_pics/person_joe.jpg",
    "comment": "Cooked a widget this morning.",
    "foodPictureUrl": "card_carrot.png",
    "timestamp": "30"
  },
  {
    "id": "5",
    "profileImageUrl": "assets/profile_pics/person_katz.jpg",
    "comment": "Going to power through my day with this nutty smoothie to finish up Flutter apprentice for our
readers!",
    "foodPictureUrl": "card_carrot.png",
    "timestamp": "40"
  },
  {
    "id": "6",
    "profileImageUrl": "assets/profile_pics/person_kevin.jpg",
    "comment": "Cooking up some steak    today, state should be rare, medium or medium well?",
    "foodPictureUrl": "card_carrot.png",
    "timestamp": "50"
  },
  {
    "id": "7",
    "profileImageUrl": "assets/profile_pics/person_sandra.jpg",
    "comment": "Kicking off the morning with mike's green smoothie recipe. Time to edit these chapters!",
    "foodPictureUrl": "card_carrot.png",
    "timestamp": "50"
  },
  {
    "id": "8",
    "profileImageUrl": "assets/profile_pics/person_manda.jpg",
    "comment": "Not sure where I should travel to eat today.",
    "foodPictureUrl": "card_carrot.png",
    "timestamp": "60"
  },
  {
    "id": "9",
    "profileImageUrl": "assets/profile_pics/person_ray.jpg",
    "comment": "Hot off the press, cooking up more books this year!",
    "foodPictureUrl": "card_carrot.png",
    "timestamp": "70"
  },
  {
    "id": "10",
    "profileImageUrl": "assets/profile_pics/person_tiffani.jpg",
    "comment": "Hmmm ... Salmon, little broccoli, mash potatos",
    "foodPictureUrl": "card_carrot.png",
    "timestamp": "90"
  }
]
}
{
  "recipes": [
    {
      "id": "1",
      "cardType": "card1",
      "title": "The Art of Dough",
      "subtitle": "Editor's Choice",
      "backgroundImage": "assets/magazine_pics/card_bread.jpg",
      "backgroundImageSource": "https://www.foodiesfeed.com/free-food-photo/baker-with-wheat-flour/",
    }
  ]
}
```

```
"message": "Learn to make the perfect bread.",
"authorName": "Ray Wenderlich",
"role": "Founder of Raywenderlich",
"authorImage": "assets/profile_pics/person_ray.jpeg",
"durationInMinutes": 15,
"dietType": "Standard",
"calories": 136,
"tags": ["Carbs", "Dough", "Crunchy", "Yummy"],
"description": "food made of flour, water, and yeast or another leavening agent, mixed together and baked: a loaf
of bread",
"source": "Apple Dictionary",
"ingredients": [
  {
    "imageUrl": "food_flour.jpg",
    "title": "Flour",
    "source": "https://pixabay.com/photos/bake-butter-flour-mountain-pile-599521/"
  },
  {
    "imageUrl": "food_dozen_eggs.jpg",
    "title": "Dozen Eggs",
    "source": "https://pixabay.com/photos/egg-hen-s-egg-food-nutrition-1374141/"
  }
],
"instructions": [
  {
    "imageUrl": "food_flour.jpg",
    "description": "Pour the flour on the table.",
    "durationInMinutes": 5
  },
  {
    "imageUrl": "food_dozen_eggs.jpg",
    "description": "Mix the eggs with the flour. ",
    "durationInMinutes": 5
  }
]
},
{
  "id": "2",
  "cardType": "card2",
  "title": "Recipe",
  "subtitle": "Smoothies",
  "backgroundImage": "assets/magazine_pics/card_smoothie.png",
  "backgroundImageSource": "https://www.foodiesfeed.com/free-food-photo/peanut-butter-and-banana-smoothie/",
  "message": "Much Tasty, Wow Smooth, Oh Nutty",
  "authorName": "Mark Ian Soriano",
  "role": "Smoothie Connoisseur",
  "profileImage": "assets/profile_pics/person_ian.jpg",
  "durationInMinutes": 15,
  "dietType": "Standard",
  "calories": 136,
  "tags": ["Drink", "Nuts", "Smooth", "Energy"],
  "description": "a thick, smooth drink of fresh fruit pureed with milk, yogurt, or ice cream.",
  "source": "Apple Dictionary",
  "ingredients": [
    {
      "imageUrl": "food_peanutbutter.png",
      "title": "Peanut Butter",
      "source": "https://pixabay.com/photos/almond-blue-cook-cooking-egg-fat-1850615/"
    },
    {
      "imageUrl": "food_soymilk.png",
      "title": "Soy Milk",
      "source": "https://pixabay.com/photos/soy-milk-soy-soybean-soy-milk-2263942/"
    }
  ]
}
```

```
    },
    {
      "imageUrl": "food_banana.jpg",
      "title": "Banana",
      "source": "https://pixabay.com/photos/bananas-fruit-food-fresh-mature-3700718/"
    }
  ],
  "instructions": [
    {
      "imageUrl": "food_banana.jpg",
      "description": "Peel the banana, cut it in half and place it in the blender.",
      "durationInMinutes": 1
    },
    {
      "imageUrl": "",
      "description": "Add two big scoops of peanut butter.",
      "durationInMinutes": 1
    },
    {
      "imageUrl": "",
      "description": "Add 2 cups of almond milk. Depending on the portion you may want to add more milk. You can add any type of milk you want.",
      "durationInMinutes": 5
    }
  ]
},
{
  "id": "3",
  "cardType": "card1",
  "title": "Delicious Salad",
  "subtitle": "Healthy Foods",
  "backgroundImage": "assets/magazine_pics/card_salad.png",
  "backgroundImageSource": "https://www.foodiesfeed.com/free-food-photo/green-salad-with-bread/",
  "message": "Learn how to make a killer salad",
  "authorName": "Kevin Moore",
  "role": "Author at RW",
  "profileImage": "assets/profile_pics/person_kevin.jpeg",
  "durationInMinutes": 15,
  "dietType": "Vegan",
  "calories": 136,
  "tags": ["Healthy", "Salads", "Lunch"],
  "description": "a cold dish of various mixtures of raw or cooked vegetables, usually seasoned with oil, vinegar, or other dressing and sometimes accompanied by meat, fish, or other ingredients: a green salad",
  "source": "Apple Dictionary",
  "ingredients": [
    {
      "imageUrl": "food_cucumber.jpg",
      "title": "Cucumber",
      "source": "https://pixabay.com/photos/cucumbers-vegetables-green-healthy-849269/"
    },
    {
      "imageUrl": "food_green_beans.jpg",
      "title": "Green beans",
      "source": "https://pixabay.com/photos/beans-vegetables-green-market-4785/"
    },
    {
      "imageUrl": "food_brussels_sprouts.jpg",
      "title": "Brussel Sprouts",
      "source": "https://pixabay.com/photos/brussels-sprouts-vegetables-sprouts-22009/"
    }
  ],
  "instructions": [
    {
```

```

    "imageUrl": "food_cucumber.jpg",
    "description": "Cut the cucumber up into many pieces.",
    "durationInMinutes": 3
  },
  {
    "imageUrl": "",
    "description": "Wash the green beans well with salt, and rinse it a couple of times.",
    "durationInMinutes": 2
  },
  {
    "imageUrl": "",
    "description": "Peel the first layer of brussel sprout leaves and place them in a bowl of water. Make sure you
add salt to kill the germs.",
    "durationInMinutes": 5
  },
  {
    "imageUrl": "",
    "description": "Place everything in a clean bowl, and add some salad dressing of your choice.",
    "durationInMinutes": 5
  }
]
},
{
  "id": "4",
  "cardType": "card3",
  "title": "Recipe Trends",
  "subtitle": "",
  "backgroundImage": "assets/magazine_pics/card_carrot.png",
  "backgroundImageSource": "https://www.foodiesfeed.com/free-food-photo/fresh-carrots-from-a-market/",
  "message": "Trends of what people are making",
  "chef": "",
  "role": "",
  "durationInMinutes": null,
  "dietType": null,
  "calories": null,
  "tags": ["Healthy", "Vegan", "Carrots", "Greens", "Wheat", "Pescetarian", "Mint", "Lemongrass", "Salad",
"Water"],
  "description": null,
  "source": null,
  "ingredients": null,
  "instructions": null
}
]
}
```