# COIMBATORE INSTITUTE OF TECHNOLOGY

(Government-Aided Autonomous Institution Affiliated To Anna University, Chennai)

COIMBATORE – 14.

## MOBILE APPLICATION DEVELOPMENT LAB-19MAM66

**SUBMITTED TO:**

**DR. D. KAVITHA**

**MS. P. HEMASHREE**

**DEPARTMENT OF COMPUTING**


**SUBMITTED BY:**

**NAME :** SARVESH PV

**REGISTER NO**: 71762234042

**DEPARTMENT:** 3rd YEAR MSc-AIML

# MEAL PLANNER


## Abstract

This project presents a dynamic Meal Planner application designed to provide users with an efficient, curated, and engaging experience for discovering meal recipes tailored to their preferences. Engineered with Flutter for cross-platform compatibility and utilizing SQLite for persistent local storage, the application seamlessly integrates modern functionalities such as API-based recipe retrieval, location-based services, secure local login management, real-time content updates via an RSS feed, and intelligent meal recommendations. The application not only supports user convenience through advanced features like categorization, favorites management, and personalized suggestions but also enriches user interaction by enabling email notifications and integrating multimedia recipe tutorials. The objective is to deliver an intuitive and sophisticated platform that elevates meal planning into a delightful and intelligent daily routine.


## Introduction

Meal planning is a critical yet often overlooked component of maintaining a balanced, healthy lifestyle. In the era of digital convenience, providing users with an intelligent, mobile-first meal planning solution significantly enhances their culinary journey. This Meal Planner app bridges the gap by offering an aesthetic, user-centric interface that retrieves high-quality recipe data from external APIs, enriched with features such as search capabilities, favorites tracking, personalized recommendations, and detailed tutorials.

Architecturally, the application leverages Flutter's robust UI toolkit combined with SQLite for local storage persistence and Riverpod for state management, ensuring a performant, reactive, and secure mobile experience. This synergy ensures both data security and exceptional responsiveness across devices.

## Modules and Descriptions

### 1. Login & Authentication Module

- **Description:** Facilitates user authentication with secure credential storage on-device.
- **Features:**
    - Local authentication via SQLite.
    - Input validation and exception handling.
    - Persistent session management within app lifecycle.
    - Triggering email notifications post-login success.
    - Redirection to the primary dashboard upon successful authentication.

**2. Recipe Retrieval and Display Module**

- **Description:** Core functionality allowing users to fetch and explore meal recipes.
- **Features:**
    - Categorized recipe browsing (e.g., Breakfast, Lunch, Desserts, etc.).
    - Real-time API integration to fetch up-to-date meal data.
    - Recipe of the Week highlight section.
    - Intelligent recommendation engine based on popular trends and user interactions.

**3. Favorites Management Module**

- **Description:** Empowers users to save, categorize, and manage their preferred recipes.
- **Features:**
    - Persistent storage of favorites within SQLite.
    - Category-wise favorites organization.
    - Add/remove favorite recipes seamlessly.
    - Offline access to saved recipes.

**4. Search Functionality**

- **Description:** Provides a swift, responsive search feature across all available recipes.
- **Features:**
    - Keyword-based dynamic search.
    - Search suggestions based on popular queries.
    - Instant retrieval and display of search results.

**5. Detailed Recipe View**

- **Description:** Displays in-depth information for each recipe.
- **Features:**
    - Ingredient listing and preparation instructions.
    - Nutritional information (where available).
    - Direct link to corresponding YouTube tutorial videos.
    - Interactive UI elements for enhanced readability.

**6. RSS Feed Integration**

- **Description:** Supplies users with the latest culinary news, trends, and new recipes.
- **Features:**
    - Periodic feed updates using background fetch (Dart Isolates for multithreading).
    - Instantaneous display of latest entries without disrupting main UI.
    - Real-time refresh to keep users abreast of new content.

**7. Location Retrieval**

- **Description:** Captures and displays the user's current geographical coordinates.
- **Features:**
    - Utilizes device GPS to fetch latitude and longitude.
    - Displays location data alongside personalized recommendations (future extensibility).

**9. Logout Module**

- **Description:** Ensures secure user session termination.
- **Features:**
    - Cleans user session data from SQLite.
    - Redirection to the login screen.
    - Prevention of unauthorized navigation post-logout.

## Methods Used

**Frontend (Flutter)**

- Dart programming language.
- Highly responsive UI crafted with stateless and stateful widgets.
- Navigator 2.0 for seamless screen transitions.
- Riverpod for robust and scalable state management.
- **Packages utilized:**
    - http for RESTful API interactions.
    - sqflite for local database operations.
    - geolocator for location services.
    - share_plus for sharing recipes.

- flutter_email_sender for email notifications.
- webfeed for RSS parsing.
- flutter_riverpod for efficient app-wide state control.

**Backend / Data Storage**

- External recipe API integration for real-time recipe data.
- SQLite database to persist login credentials, favorites, and user preferences.
- RESTful API standards for secure and structured data fetching.

**Multithreading (Dart Isolates)**

- Employed to offload RSS feed parsing and updates.
- Guarantees uninterrupted UI responsiveness during heavy background operations.
- Concurrent retrieval and rendering of new feed entries.

**Email Notification**

- Post-login email confirmation via flutter_email_sender.
- Configurable templates for user engagement and security acknowledgment.

**Location Services**

- Fetches device coordinates using geolocator.
- Future scope for location-based recipe recommendations or event-specific meals.

## Conclusion

The Meal Planner App exemplifies the successful integration of cutting-edge mobile development practices to create a comprehensive, aesthetically pleasing, and highly functional meal discovery platform. By synergizing Flutter’s modern UI capabilities, Riverpod’s scalable state management, and SQLite’s local persistence, the application offers users an intelligent and deeply personalized meal planning experience. Features such as recipe recommendations, multimedia tutorial integration, real-time RSS updates, and location awareness collectively enhance user engagement, usability, and retention.

Anticipated future upgrades include calendar-based meal scheduling, personalized nutrition tracking, AI-based smart meal suggestions, and voice command integrations, further cementing Meal Planner's position as a next-generation lifestyle application.

## Code Snippets

## 1.Database

```dart
// Initialize the local database
Future<void> _initializeDB() async {
  final dbPath = await getDatabasesPath();
  final path = join(dbPath, 'mealplanner.db');
  _database = await openDatabase(path, version: 1, onCreate: (db, version) {
    return db.execute(
      'CREATE TABLE users(id INTEGER PRIMARY KEY, name TEXT, email TEXT, login_time TEXT)',
    );
  });
}

// Save user details in both the database and GetStorage
Future<void> _saveUser(String name, String email) async {
  final timestamp = DateFormat.yMd().add_jm().format(DateTime.now());
  await _database?.insert('users', {
    'name': name,
    'email': email,
    'login_time': timestamp,
  });

  // Save to GetStorage for persistence
  await _storage.write('user_name', name);
  await _storage.write('user_email', email);
  await _storage.write('login_time', timestamp);

  // Send welcome email
  await _sendEmail(email, timestamp);
}
```

## 2. E-Mail

```dart
// Send a welcome email to the user
Future<void> _sendEmail(String email, String timestamp) async {
  const username = 'sammie251981@gmail.com';
  const password = 'qmbz gida yjlt wflr';
  final smtpServer = gmail(username, password);

  final message = Message()
    ..from = const Address(username, 'Meal Planner')
    ..recipients.add(email)
    ..subject = 'Login Successful'
    ..text =
        'Hi $userName!\n\nYou logged in on $timestamp.\n\nBon appétit! 🍽️';

  try {
    await send(message, smtpServer);
  } catch (e) {
    debugPrint("Failed to send email: $e");
  }
}
```

### 3. Location Service

```dart
// Fetch the current location of the user
Future<Position> getLocation() async {
  bool serviceEnabled = await Geolocator.isLocationServiceEnabled();
  if (!serviceEnabled) throw Exception('Location services are disabled.');
  LocationPermission permission = await Geolocator.checkPermission();
  if (permission == LocationPermission.denied) {
    permission = await Geolocator.requestPermission();
    throw Exception('Location permissions are denied.');
  }
  if (permission == LocationPermission.deniedForever) {
    throw Exception('Location permissions are permanently denied.');
  }
  return await Geolocator.getCurrentPosition();
}
```

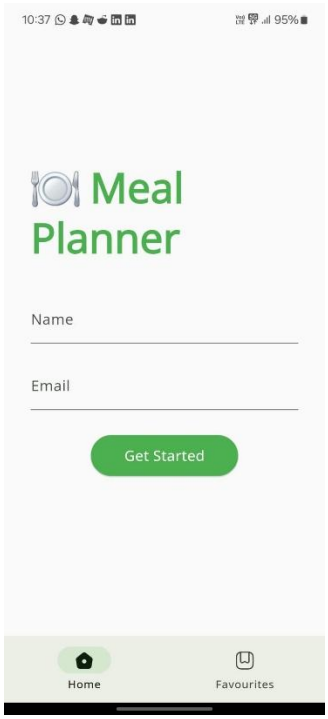
### 4. RSS Feed

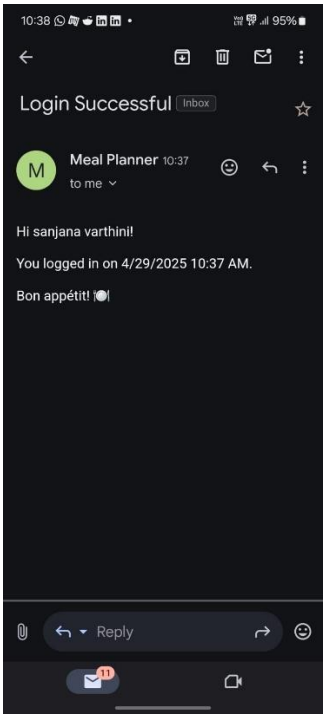
```dart
// Fetch RSS feed data
Future<RssFeed?> fetchRssFeed() async {
  final response =
      await http.get(Uri.parse('https://www.bbcgoodfood.com/rss'));
  if (response.statusCode == 200) {
    return RssFeed.parse(response.body);
  }
  return null;
}
```

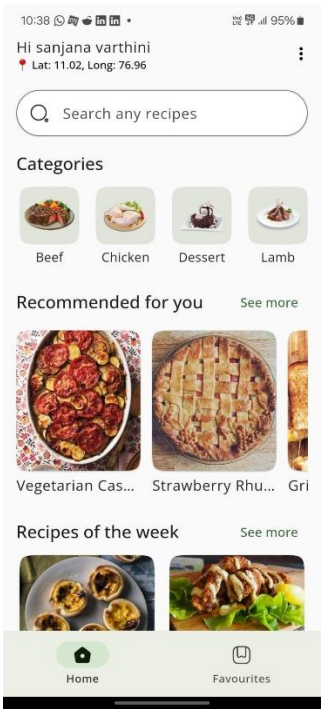
### 5. Search

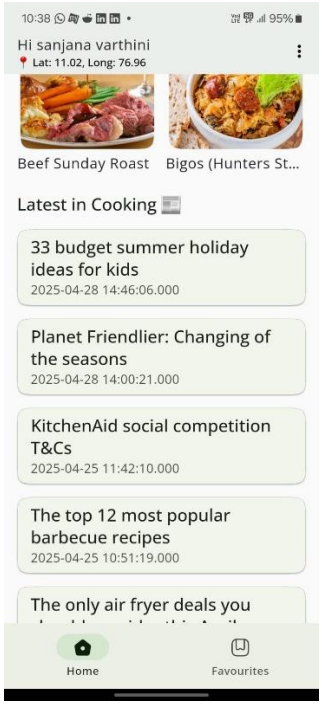
```dart
@riverpod
Future<List<CompleteRecipe>> searchRecipes(SearchRecipesRef ref, String query) async {
  final url = "https://themealdb.com/api/json/v1/1/search.php?s=$query";
  final response = await dio.get(url);
  if (response.statusCode == 200) {
    final data = response.data['meals'];
    return List.from(data).map((e) => CompleteRecipe.fromMap(e)).toList();
  }
  return [];
}
```

# Output :


Meal Planner
Name
Email
Get Started
Home
Favourites

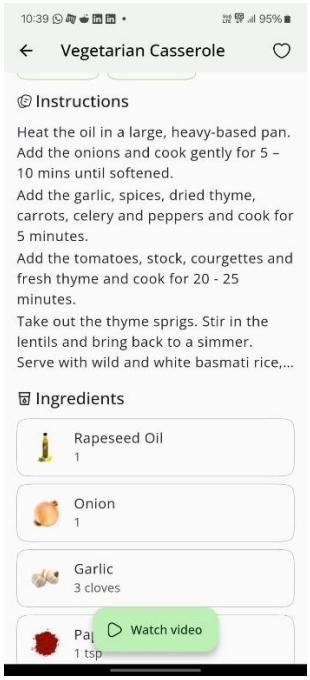

Login Successful
Inbox
Meal Planner 10:37
to me
Hi sanjana varthini!
You logged in on 4/29/2025 10:37 AM.
Bon appétit!
Reply

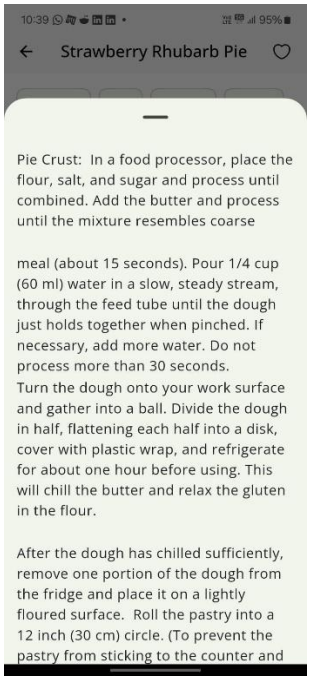

Hi sanjana varthini
Lat: 11.02, Long: 76.96
Search any recipes
Categories
Beef
Chicken
Dessert
Lamb
Recommended for you
See more
Vegetarian Cas...
Strawberry Rhu...
Recipes of the week
See more
Home
Favourites

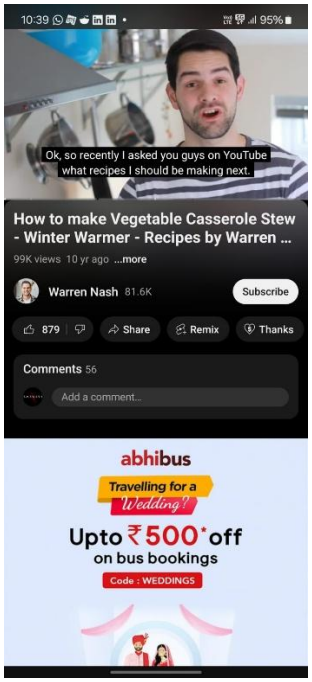

Hi sanjana varthini
Lat: 11.02, Long: 76.96
Beef Sunday Roast
Bigos (Hunters St...
Latest in Cooking
33 budget summer holiday ideas for kids
2025-04-28 14:46:06.000
Planet Friendlier: Changing of the seasons
2025-04-28 14:00:21.000
KitchenAid social competition T&Cs
2025-04-25 11:42:10.000
The top 12 most popular barbecue recipes
2025-04-25 10:51:19.000
The only air fryer deals you
Home
Favourites

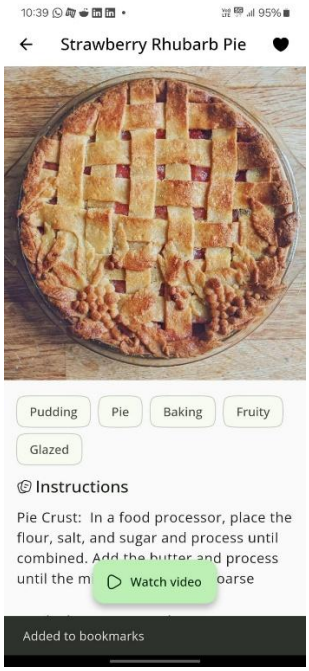

Hi sanjana varthini
Lat: 11.02, Long: 76.96
Beef Sunday Roast
Bigos (Hunters St...
Latest in Cooking
KitchenAid social competition T&Cs
Find out how to enter our competition to win a KitchenAid 4.8L Artisan Stand Mixer in Pistachio plus a copy of 'The Frozen Peas Cookbook' by Sam Goldsmith
Close
The top 12 most popular barbecue recipes
2025-04-25 10:51:19.000
The only air fryer deals you
Home
Favourites


Vegetarian Casserole
Casserole
Vegetarian
Instructions
Heat the oil in a large, heavy-based pan. Add the onions and cook gently for 5 – 10 mins until softened.
Add the garlic, spices, dried thyme, carrots, celery and peppers and cook for 5 minutes.
Watch video
minutes.

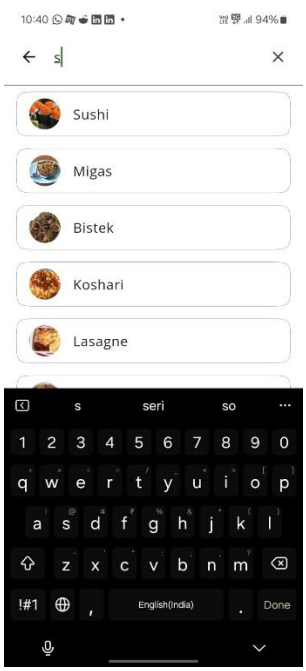

Vegetarian Casserole
Instructions
Heat the oil in a large, heavy-based pan. Add the onions and cook gently for 5 – 10 mins until softened.
Add the garlic, spices, dried thyme, carrots, celery and peppers and cook for 5 minutes.
Add the tomatoes, stock, courgettes and fresh thyme and cook for 20 - 25 minutes.
Take out the thyme sprigs. Stir in the lentils and bring back to a simmer.
Serve with wild and white basmati rice,...
Ingredients
Rapeseed Oil
1
Onion
1
Garlic
3 cloves
1 tsp
Watch video


Strawberry Rhubarb Pie
Pie Crust: In a food processor, place the flour, salt, and sugar and process until combined. Add the butter and process until the mixture resembles coarse
meal (about 15 seconds). Pour 1/4 cup (60 ml) water in a slow, steady stream, through the feed tube until the dough just holds together when pinched. If necessary, add more water. Do not process more than 30 seconds.
Turn the dough onto your work surface and gather into a ball. Divide the dough in half, flattening each half into a disk, cover with plastic wrap, and refrigerate for about one hour before using. This will chill the butter and relax the gluten in the flour.
After the dough has chilled sufficiently, remove one portion of the dough from the fridge and place it on a lightly floured surface. Roll the pastry into a 12 inch (30 cm) circle. (To prevent the pastry from sticking to the counter and


Ok, so recently I asked you guys on YouTube what recipes I should be making next.
How to make Vegetable Casserole Stew - Winter Warmer - Recipes by Warren ...
99K views 10 yr ago ...more
Warren Nash 81.6K
Subscribe
879
Share
Remix
Thanks
Comments 56
Add a comment...
abhibus
Travelling for a Wedding?
Upto ₹500* off
on bus bookings
Code : WEDDING5


Strawberry Rhubarb Pie
Pudding
Pie
Baking
Fruity
Glazed
Instructions
Pie Crust: In a food processor, place the flour, salt, and sugar and process until combined. Add the butter and process until the m
oarse
Watch video
Added to bookmarks


Favourite Recipes
Dessert
See more
Home
Favourites


s
Sushi
Migas
Bistek
Koshari
Lasagne
seri
so
Done
English(India)