SP-5 — Purple Groceries App

PROJECT DEVELOPMENT DOC

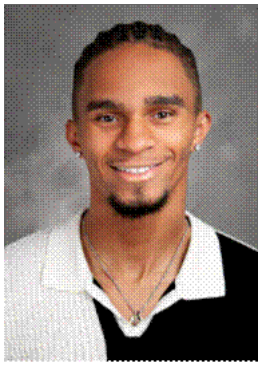
Course: CS 4850 Senior Project

Section 03

Semester: Fall 2025

Date: October 26, 2025

Professor: Sharon Perry

			
Carinne Tzurdecker Documentation	Vanessa Espinoza Documentation	Shane Hazeur Developer	Piao Hou Developer

---

Table of Contents

---

1. Development Summary

- Frontend
    - **Frameworks/Libraries Used:** Flutter, Dart
    - **UI/UX Design Choices:** Material Design 3, Responsive layout, Light/Dark mode, Minimal color palette
    - **Key Components Developed:** Recipe Cards, Grocery List Items, Meal Category Tiles, Add/Edit Item Form, Navigation Drawer
- Backend
    - **APIs Implemented:** Firebase (planned), Local JSON storage, HTTP endpoints (for testing)
    - **Authentication/Authorization:** Firebase Auth
    - **Core Logic or Algorithms:** Ingredient-based recipe filtering, Dynamic grocery list updates, Category sorting
- Integration
    - **Communication Between Frontend and Backend:** HTTP requests (Dio package), Firebase SDK
    - **API Endpoints:** /recipes, /grocery-items, /auth/login, /auth/register

- Special Features
    - AI Responses
    - Recipe/Grocery Planner

Frontend

> The frontend was developed using Flutter and Dart for cross-platform compatibility across Android and desktop. The design follows Material Design 3 standards with a clean, minimal interface. Components like recipe cards, grocery list items, and meal tiles were built as reusable widgets to support scalability and smooth navigation.

Backend

> Firebase handles user authentication and cloud data storage. The backend connects through the Firebase SDK for real-time updates to grocery items and recipe data. Each user's lists sync automatically across sessions using state listeners.

---

2. Database Connection

Purpose:

Data is stored locally within the app's storage, not in any external database or cloud service. Used to persist lightweight app data such as authentication tokens, user info, and preferences.

Connection Details:

- *No server-side or remote connection.*
- *Data stored locally on the device using key-value pairs.*
- *No data synchronization between devices.*
- *Data is cleared if the app is uninstalled.*

Authentication:

- *Mock authentication only (no backend).*
- *Validates only input format (e.g., email pattern, password length).*
- *Generates placeholder tokens like mock_jwt_token.*
- *No real user verification or API calls.*

ORM / Query Tool:

Not applicable — uses shared_preferences for storage.

Storage Technology:

- ***Package:** shared_preferences: ^2.2.2*
- ***Type:** Local key-value storage*
- ***Stored Values:***
    - *auth_token*
    - *user_id, user_email*
    - *gemini_api_key*
    - *app_preferences*

---

3. How to Set Up the Project

1. Install Flutter
    - Download Flutter from: https://flutter.dev/docs/get-started/install/windows
    - Follow the installation instructions for your OS
    - Verify installation by running: flutter doctor
    - Ensure Chrome is installed (comes with most Windows systems)

2. Install Git (if not already installed)

- Download from: https://git-scm.com/downloads
- Follow installation wizard

Step-by-Step Setup

- Step 1: Clone the Repository

  git clone <repository-url>
  cd recipe_meal_planner
  Note: Replace <repository-url> with your actual GitHub repository URL.

- Step 2: Navigate to Project Directory

  Open PowerShell or Command Prompt and navigate to the project:
  cd "C:\Users\YourName\recipe_meal_planner"

- Step 3: Install Dependencies

  flutter pub get
  This will download all required packages listed in pubspec.yaml.

- Step 4: Get Your FREE Gemini API Key
  1. Go to: https://gemini.google.com/apps/apikey (Or use this one for now: AIzaSyCSHGLIyidOK-RfatiJhW3X0tG4_0itQoY)
  2. Sign in with your Google account
  3. Click "Create API Key" or "Get API Key"
  4. Copy the generated API key (save it somewhere safe - you'll need it later)
- Step 5: Run the App

  flutter run -d chrome
  First time running might take 2-3 minutes while Flutter builds the app.
  Note: If you get an error about "build directory locked":
  - Close all Chrome windows
  - Open Task Manager (Ctrl+Shift+Esc)
  - End any chrome.exe or dart.exe processes
  - Run flutter run -d chrome again
- Step 6: Configure Gemini API Key
  1. The app should open in Chrome automatically
  2. Navigate to Settings (gear icon)
  3. Click "AI Chat Settings"
  4. Paste your Gemini API key

5. Click "Save"

- Step 7: Test the AI Chat Feature
    1. Navigate to Chat Brief in the app
    2. Type a message like: "I want healthy vegetarian meals"
    3. Press Send
    4. You should receive an AI-powered response from Gemini!

Troubleshooting

- Issue: "No pubspec.yaml file found"
- Solution: Make sure you're in the project root directory. Run:

```
cd recipe_meal_planner
```

- Issue: Build directory locked error
- Solution:
    1. Close Chrome completely
    2. Run: flutter clean
    3. Run: flutter run -d chrome
- Issue: Gemini API not working
- Solution:
    1. Verify your API key is correct at https://gemini.google.com/apps/apikey
    2. Make sure you saved it in Settings → AI Chat Settings
    3. Restart the app after saving the key
- Issue: Dependencies not installing
- Solution:

```
flutter clean
flutter pub cache repair
flutter pub get
```

- Issue: Chrome not launching
- Solution:
    - Make sure Chrome is installed
    - Try: flutter devices to see available devices
    - Use: flutter run -d web-server --web-port=8080 to run on a different port