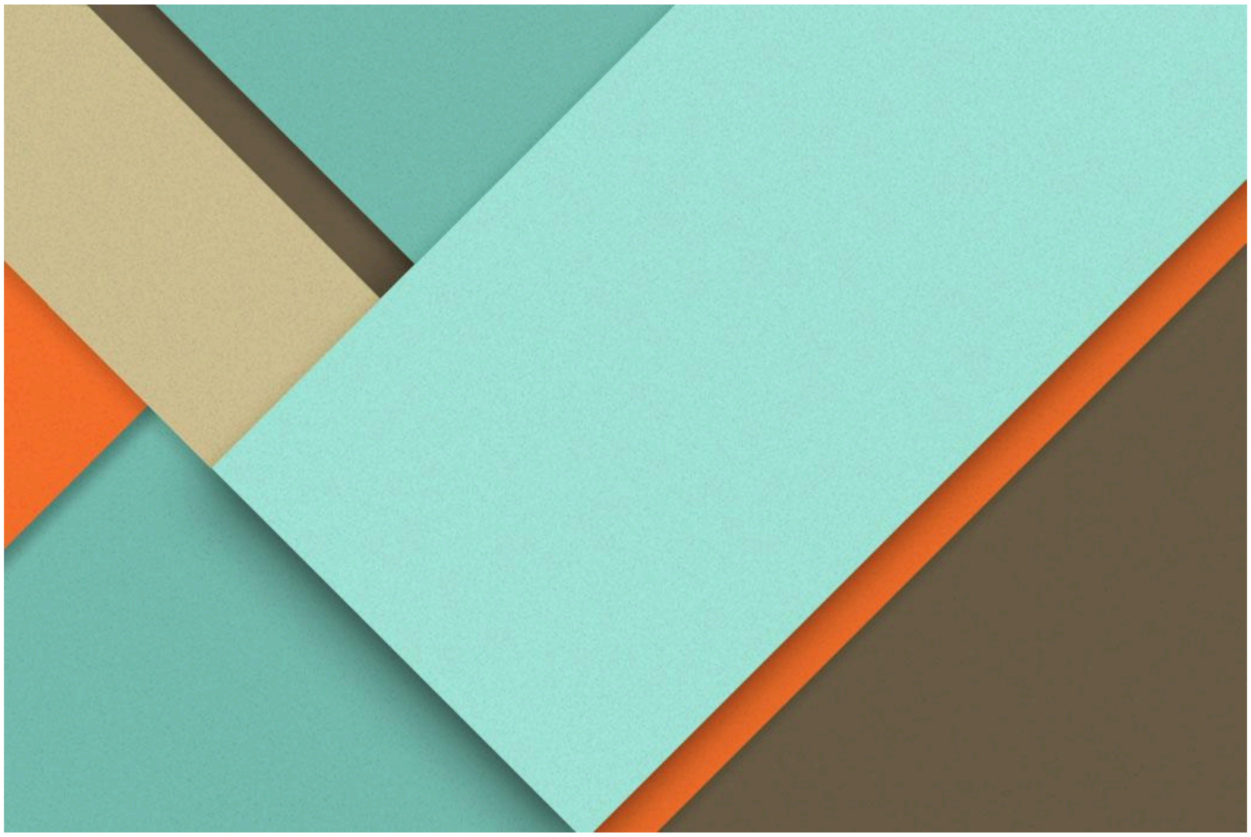


Pizza Corner

CEP (Mobile Application Development)

Submitted by:

Varoon Kumar (22SW035)

Muhammad Sajid (22SW053)

Submitted To:

Ms. Mariam Memon

Real-World Problem Identification

In the modern competitive culinary market, customers demand a fast, user-friendly, and reliable way to order food, especially on mobile devices. Many small to medium-sized restaurants lack a dedicated, high-performance mobile application, relying instead on cumbersome web portals or third-party aggregator services.

The primary challenges addressed by this project include:

- **Poor Mobile Experience:** Web-based ordering systems are often slow and not optimized for touch interfaces.
- **Lack of Offline Persistence:** A critical need for local storage to maintain the user's cart state during network interruptions or app restarts.
- **Complex Ordering Flow:** Ordering apps must simplify selection, customization (like size/quantity), and checkout to minimize abandonment.
- **Store Management Gap (NEW):** Restaurants need a straightforward interface to view and manage incoming orders in real-time, which is currently missing in simple client-side ordering apps.

Proposed Solution: The Flutter Pizza App

We have developed a cross-platform Pizza Ordering App using Flutter for a beautiful, responsive frontend and SQFlite for reliable local data persistence. The application serves two main user types: the Customer (for placing orders) and the Shop Owner (via the Admin Panel).

Key Features

I. Interactive Dashboard:

An interactive dashboard serves as the main hub for users, allowing customers to browse featured pizzas, view deals, and access recent orders instantly.

II. Customization Option:

Customers can easily select their desired pizza size (Small, Medium, Large), choose from available toppings, and adjust quantity. The total price automatically updates in real-time based on user selections.

III. Checkout & Order Placement:

Collects customer details and moves cart contents to the `orders` and `order_items` tables upon successful checkout.

IV. Offline Menu Access:

Menu items are pre-populated and stored locally in the SQFlite database for instant access (`database_helper.dart`).

V. Admin Panel:

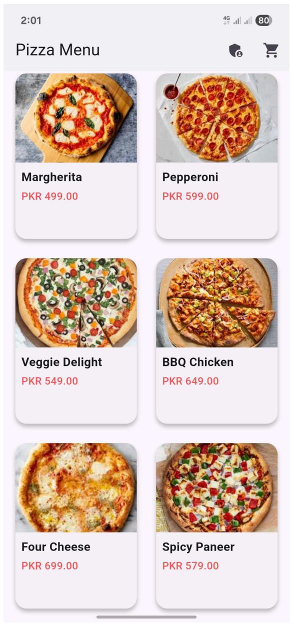
A dedicated screen for the shop owner to monitor all incoming orders and their details.

Screens & Functionality

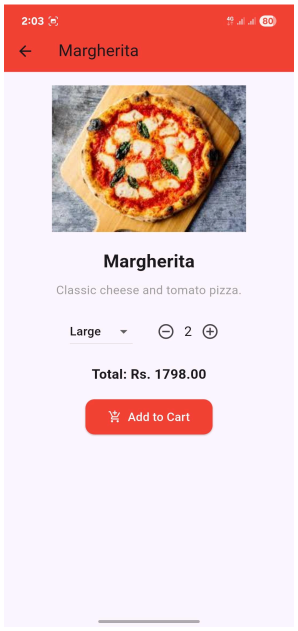
The app's UI is built on Flutter's Material Design principles, providing a clean and intuitive ordering experience.

Key Screens

I. Home Screen:



II. Order Pizza Screen:



2:03

Margherita

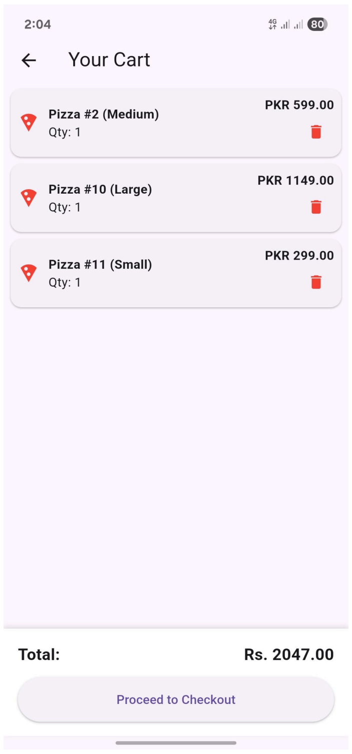
Margherita

Classic cheese and tomato pizza.

Large 2

Total: Rs. 1798.00

III. Cart Screen:



2:04

Your Cart

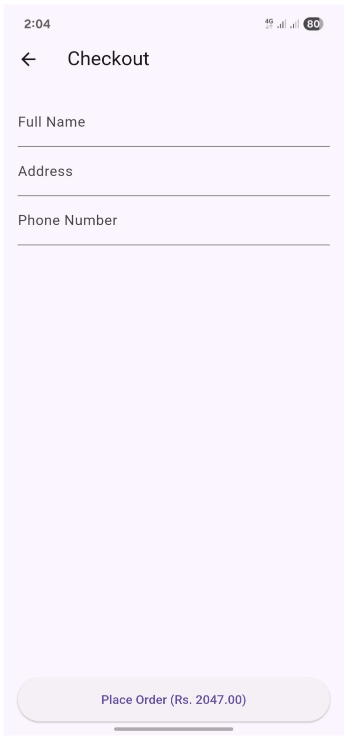
Pizza #2 (Medium) — Qty: 1 — PKR 599.00

Pizza #10 (Large) — Qty: 1 — PKR 1149.00

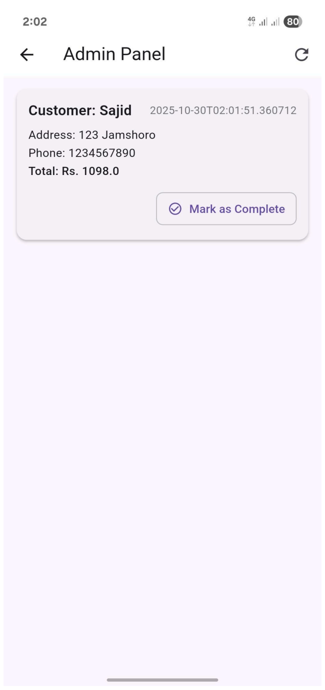
Pizza #11 (Small) — Qty: 1 — PKR 299.00

Total: Rs. 2047.00

IV. Checkout Screen:



V. Admin Order View:



Customer: Sajid 2025-10-30T02:01:51.360712
Address: 123 Jamshoro
Phone: 1234567890
Total: Rs. 1098.0

Data Storage and SQFlite Implementation

The project relies on SQFlite for reliable, structured, local data storage for the menu and all transactional records.

Table	Admin Panel Use	Key Fields Used
orders	**Primary Source.** The Admin Panel uses the `getOrders()` function to list all orders placed.	`id`, `customer_name`, `total_amount`, `order_date`
order_items	Used to view the **details** (e.g., "1 Large Pepperoni") when the admin taps a specific order record.	`order_id`, `pizza_name`, `size`, `quantity`

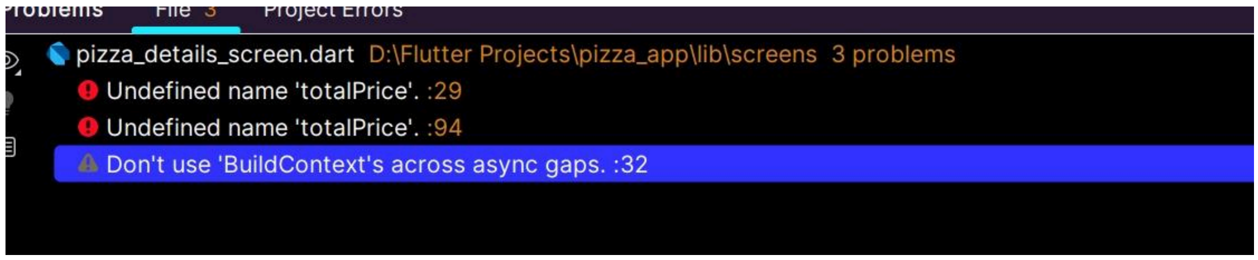
Justification

By storing all order information (`orders` and `order_items`) locally, the **Admin Panel** can function by simply querying this data, making it a powerful, self-contained demonstration of a full transaction lifecycle on the device.

Issues and Bugs Encountered and Resolved

The app's UI is built on Flutter's Material Design principles, providing a clean and intuitive ordering experience.

I. Issue #1: Undefined Variable Reference:

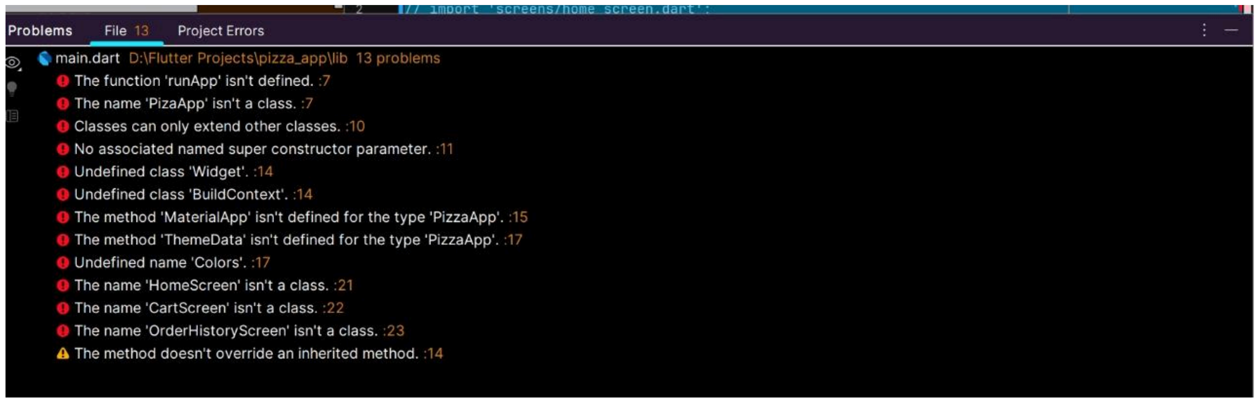


```
Problems    File 3    Project Errors
pizza_details_screen.dart  D:\Flutter Projects\pizza_app\lib\screens  3 problems
    Undefined name 'totalPrice'. :29
    Undefined name 'totalPrice'. :94
    Don't use 'BuildContext's across async gaps. :32
```

RESOLUTION:

Define the variable `totalPrice` within the scope of the class or method where it is being used, ensuring it is correctly spelled and initialized.

II. Issue #2: Critical Dependency Failure:

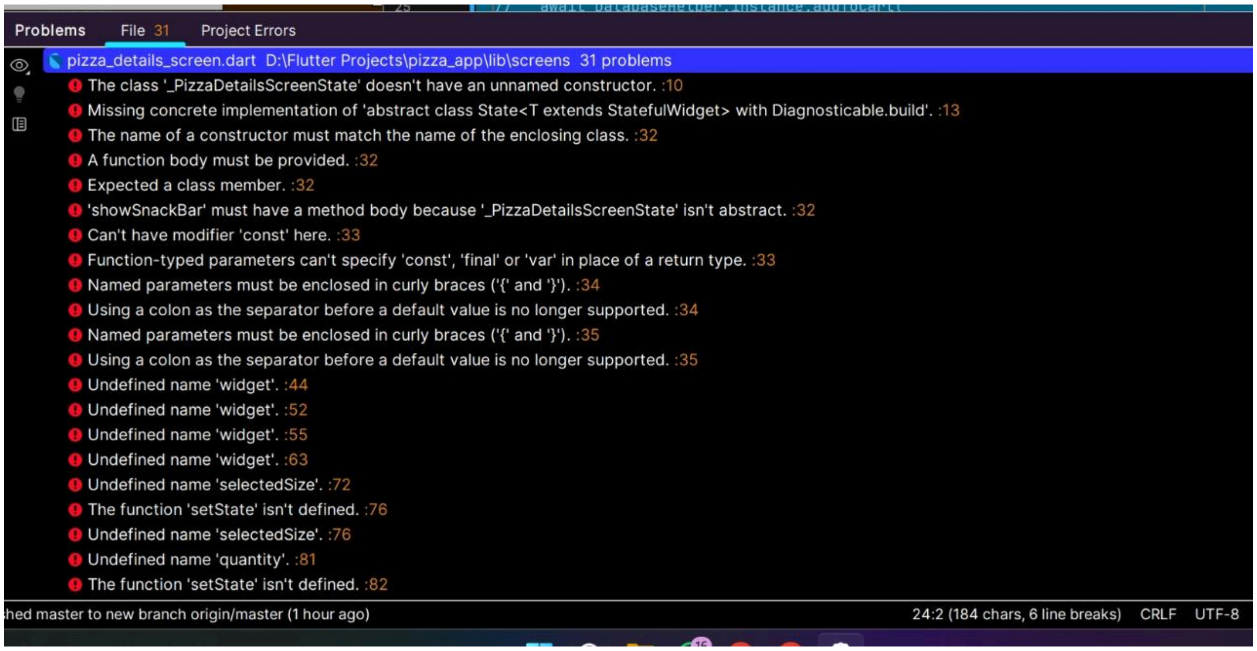


```
Problems    File 13    Project Errors
main.dart  D:\Flutter Projects\pizza_app\lib  13 problems
    The function 'runApp' isn't defined. :7
    The name 'PizaApp' isn't a class. :7
    Classes can only extend other classes. :10
    No associated named super constructor parameter. :11
    Undefined class 'Widget'. :14
    Undefined class 'BuildContext'. :14
    The method 'MaterialApp' isn't defined for the type 'PizzaApp'. :15
    The method 'ThemeData' isn't defined for the type 'PizzaApp'. :17
    Undefined name 'Colors'. :17
    The name 'HomeScreen' isn't a class. :21
    The name 'CartScreen' isn't a class. :22
    The name 'OrderHistoryScreen' isn't a class. :23
    The method doesn't override an inherited method. :14
```

RESOLUTION:

Insert the essential `import 'package:flutter/material.dart';` statement at the beginning of the `main.dart` file. This action will define the core Flutter components and widgets, resolving the bulk of the compilation errors.

III. Issue #3: Structural and Syntax Errors in PizzaDetailsScreen State Implementation:

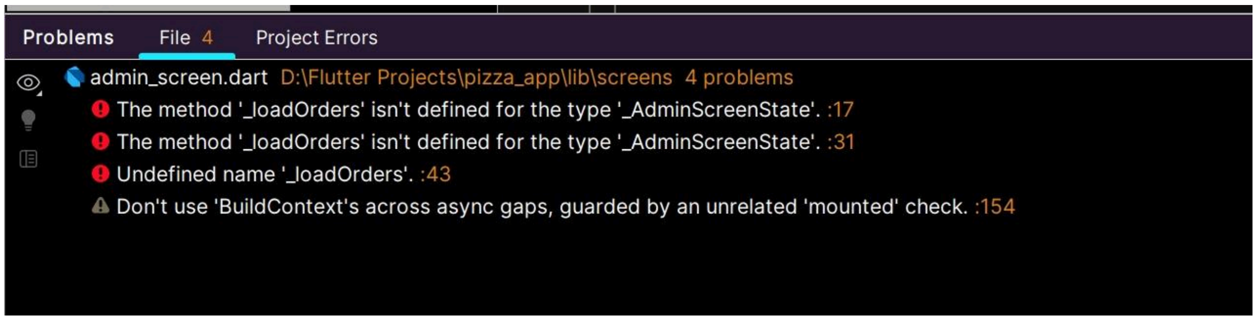


```
Problems   File 31   Project Errors
pizza_details_screen.dart  D:\Flutter Projects\pizza_app\lib\screens  31 problems
  The class '_PizzaDetailsScreenState' doesn't have an unnamed constructor. :10
  Missing concrete implementation of 'abstract class State<T extends StatefulWidget> with Diagnosticable.build'. :13
  The name of a constructor must match the name of the enclosing class. :32
  A function body must be provided. :32
  Expected a class member. :32
  'showSnackBar' must have a method body because '_PizzaDetailsScreenState' isn't abstract. :32
  Can't have modifier 'const' here. :33
  Function-typed parameters can't specify 'const', 'final' or 'var' in place of a return type. :33
  Named parameters must be enclosed in curly braces ('{' and '}'). :34
  Using a colon as the separator before a default value is no longer supported. :34
  Named parameters must be enclosed in curly braces ('{' and '}'). :35
  Using a colon as the separator before a default value is no longer supported. :35
  Undefined name 'widget'. :44
  Undefined name 'widget'. :52
  Undefined name 'widget'. :55
  Undefined name 'widget'. :63
  Undefined name 'selectedSize'. :72
  The function 'setState' isn't defined. :76
  Undefined name 'selectedSize'. :76
  Undefined name 'quantity'. :81
  The function 'setState' isn't defined. :82
hed master to new branch origin/master (1 hour ago)          24:2 (184 chars, 6 line breaks)   CRLF   UTF-8
```

RESOLUTION:

Review and correct the fundamental structure of the `PizzaDetailsScreen` and its associated **`State`** class to ensure proper inheritance and implementation of methods like **`createState()`** and **`build()`**. Additionally, correct the syntax for named function parameters by using **`{...}`** and the **`=`** sign for default values.

IV. Issue #4: Undefined Method _loadOrders and Asynchronous Context Warning:



```
Problems   File 4   Project Errors
admin_screen.dart  D:\Flutter Projects\pizza_app\lib\screens  4 problems
  The method '_loadOrders' isn't defined for the type '_AdminScreenState'. :17
  The method '_loadOrders' isn't defined for the type '_AdminScreenState'. :31
  Undefined name '_loadOrders'. :43
  Don't use 'BuildContext's across async gaps, guarded by an unrelated 'mounted' check. :154
```

RESOLUTION:

Implement the missing **`_loadOrders`** method within the `_AdminScreenState` class. Correct the warning by verifying that the **`mounted`** check directly guards the use of **`BuildContext`** after the asynchronous operation has completed.

Github Link: https://github.com/varoonk21/Pizza-Corner-App