



RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054
(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

Internship Report

on

Mobile Application Development

INT410: Intra Institutional Internship

STUDENT NAME : SRUJAN M.P

USN : 1MS24CS189

STUDENT NAME : SYED ATEEF

USN : 1MS24CS195

Ramaiah Institute of Technology

(Autonomous Institute, Affiliated to VTU)

MSR Nagar, MSRIT Post, Bangalore-560054

August - 2025



RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054
(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that Mr./Ms. Srujan M.P and Syed Ateef, the students of Bachelor of Engineering, bearing USNs: 1MS24CS189 and 1MS24CS195 has successfully completed, 24 Hours: from 05.08.2025 to 12.08.2025 Intra Institutional Internship in Mobile Application Development from the Department of Computer Science & Engineering, M S Ramaiah Institute of Technology, Bangalore.

SL No.	Component	Maximum Marks	Marks Obtained
1	Continuous Evaluation	50	
2	Presentation	20	
3	Report	30	
Total Marks		100	

Signature of the Student with Date

Signature of the Faculty Co-Ordinator

Signature of Head of the Department

OVERVIEW OF INTERNSHIP ACTIVITIES

DATE	DAY	NAME OF THE TOPIC COMPLETED
05/08/2025	Tuesday	Flutter application download and basic flutter programs
06/08/2025	Wednesday	Flutter application download and basic dart programs
07/08/2025	Thursday	Dart programs with functions and assignment
09/08/2025	Saturday	Animation in flutter and assignment
11/08/2025	Monday	Test on Dart programming and Flutter and basics of firebase
12/08/2025	Tuesday	Firebase and Demo on Project

TABLE OF CONTENTS

Contents	Page No.
1. Overall view of the project in terms of implementation	04-05
2. Code of main Modules	05-07
3. Result Snapshots	07-12
4. Conclusion	12-13

1. Overall View of the Project in Terms of Implementation

The CampusSwap app is a full-featured prototype of a campus-specific e-commerce platform. It is implemented using the **Flutter framework**, which allows for a single codebase to build a native-like experience on both iOS and Android. The app's architecture is organized into distinct, manageable parts: a main state management widget, several data models, and a collection of dedicated screen widgets for each major feature.

The app's **state management** is handled at the top level by the `_CampusSwapAppState` widget. This central widget holds the core data, such as the lists of items, placed orders, and user profiles. This approach, while simple and effective for a small-scale app, is a form of inherited state management.

Key implementations include:

- **Authentication:** The app simulates user login and registration without a backend. It stores user accounts in a simple list within the app's state.
- **Data Models:** Clear data models (Item, Order, UserProfile, etc.) are defined to structure the data, making the code readable and organized.
- **UI/UX:** The user interface is designed with a clean, material-style look. It uses a Drawer for navigation, Card widgets for item listings, and standard form elements for user input.
- **Image Handling:** The app uses the `file_picker` package to allow users to upload images for their items. These images are stored as `Uint8List` in memory for the duration of the session, simulating image storage.
- **Feature Integration:** The project successfully integrates several core e-commerce functionalities, including product Browse, a shopping cart, an order system, and a feedback mechanism.

The implementation is modular and easy to understand, with each screen widget handling its own specific logic and UI. This makes the codebase maintainable and scalable for future development.

2. Code of Main Modules

Here is a look at the code for the most important modules, demonstrating their structure and purpose.

App Root (CampusSwapApp)

This is the entry point of the application and manages the overall state and navigation flow.

Dart

```
class _CampusSwapAppState extends State<CampusSwapApp> {

  final List<Order> _placedOrders = [];

  final List<ProductFeedback> _productFeedback = [];

  UserProfile? _currentUserProfile;

  final List<UserAccount> _users = [];

  final List<Item> _items = [

    // ... initial items ...

  ];

  // Functions to manage state (e.g., _login, _logout, _addOrder)

  void _login(UserProfile userProfile) => setState(() => _currentUserProfile = userProfile);

  // ... other state management functions ...

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: _currentUserProfile == null

        ? LoginScreen( /* ... */ )

        : HomeScreen( /* ... */ ),

    );

  }

}
```

}

Login Screen (LoginScreen)

This module handles user authentication. It uses a Form with TextFormFields for user input and includes logic to validate credentials against the stored user data. It also provides a link to the account creation page.

Dart

```
class _LoginScreenState extends State<LoginScreen> {
  final _formKey = GlobalKey<FormState>();
  String _email = "";
  String _password = "";

  void _submitLogin() {
    if (_formKey.currentState!.validate()) {
      _formKey.currentState!.save();
      final profile = widget.authenticate(_email.trim(), _password.trim());
      if (profile == null) {
        // Show error
      } else {
        widget.onLogin(profile);
      }
    }
  }
}

@override
Widget build(BuildContext context) {
  // ... UI with Form and TextFormFields ...
}
```

```
}
```

Home Screen (HomeScreen)

This module is the main hub of the app. It displays the list of items for sale, with each item shown in a Card. It also includes an AppBar and a Drawer for navigation.

Dart

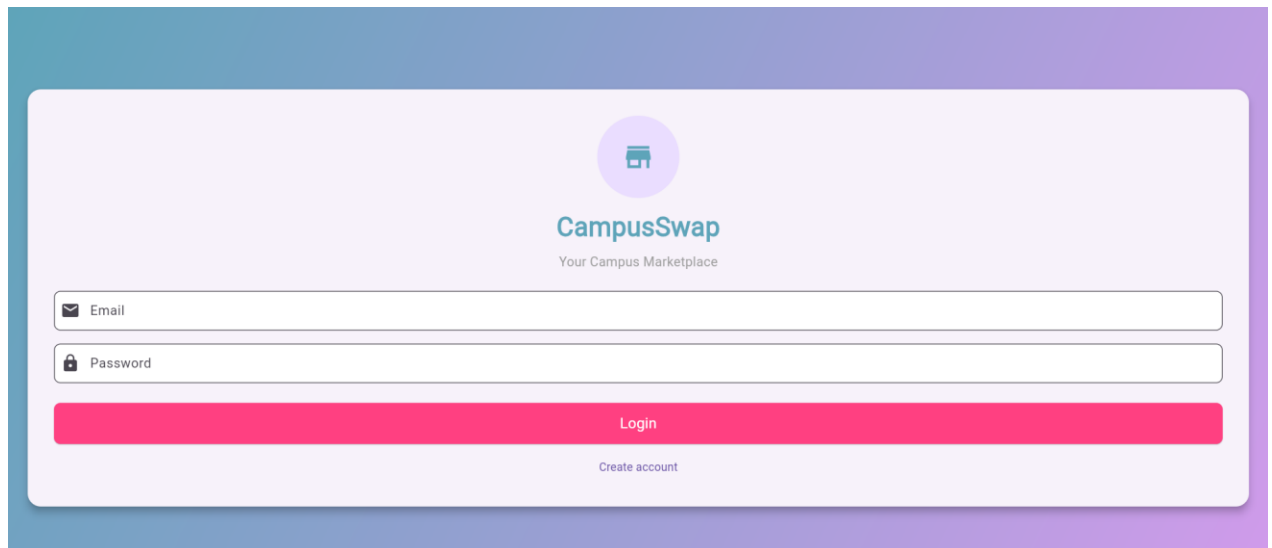
```
class _HomeScreenState extends State<HomeScreen> {  
  
  // ... state for cart items ...  
  
  @override  
  
  Widget build(BuildContext context) {  
  
    return Scaffold(  
  
      appBar: AppBar( /* ... */ ),  
  
      drawer: Drawer( /* ... */ ),  
  
      body: ListView.builder(  
  
        itemCount: widget.items.length,  
  
        itemBuilder: (ctx, i) {  
  
          final item = widget.items[i];  
  
          // ... UI for each item using Card and ListTile ...  
  
        },  
  
      ),  
  
      floatingActionButton: FloatingActionButton(onPressed: _openAddItemScreen, child: const  
Icon(Icons.add)),  
  
    );  
  
  }  
  
}
```


3. Result Snapshots

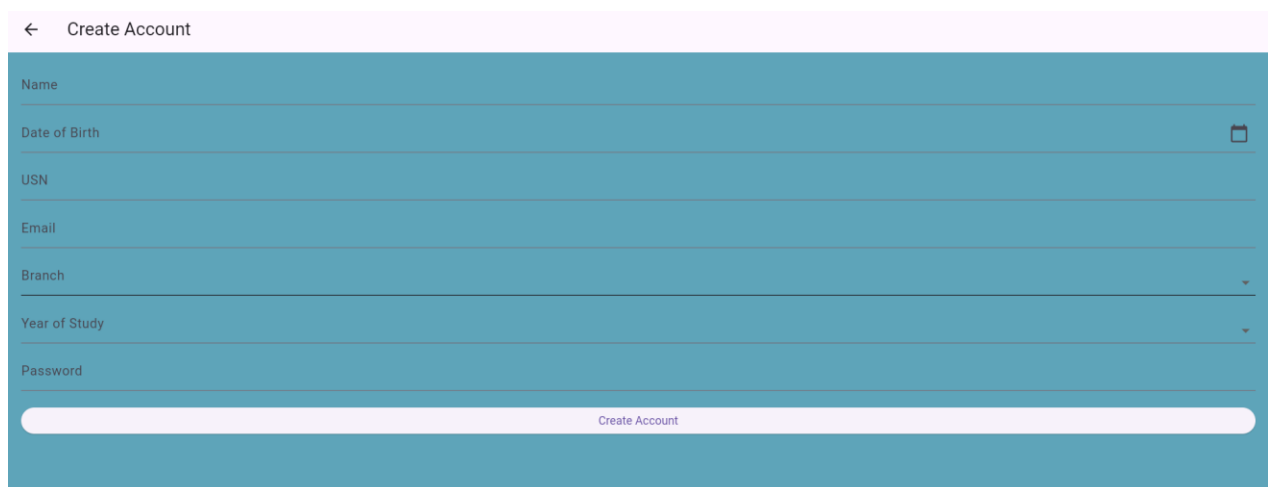
Here are some conceptual snapshots of what the user interface looks like at different stages of the application.

Login Screen

This is the first screen the user sees, featuring a welcoming design with the app logo and input fields for email and password. It also has a "Create account" button



The login screen features a light purple background with a white rounded rectangle in the center. At the top of the rectangle is the CampusSwap logo, which consists of a purple circle with a white building icon inside, followed by the text "CampusSwap" in a bold, dark blue font, and "Your Campus Marketplace" in a smaller, lighter blue font below it. Below the logo are two input fields: the first is labeled "Email" with an envelope icon on the left, and the second is labeled "Password" with a lock icon on the left. Below these fields is a large, bright pink button labeled "Login". At the bottom of the white rectangle, centered, is a link labeled "Create account" in a small, dark blue font.

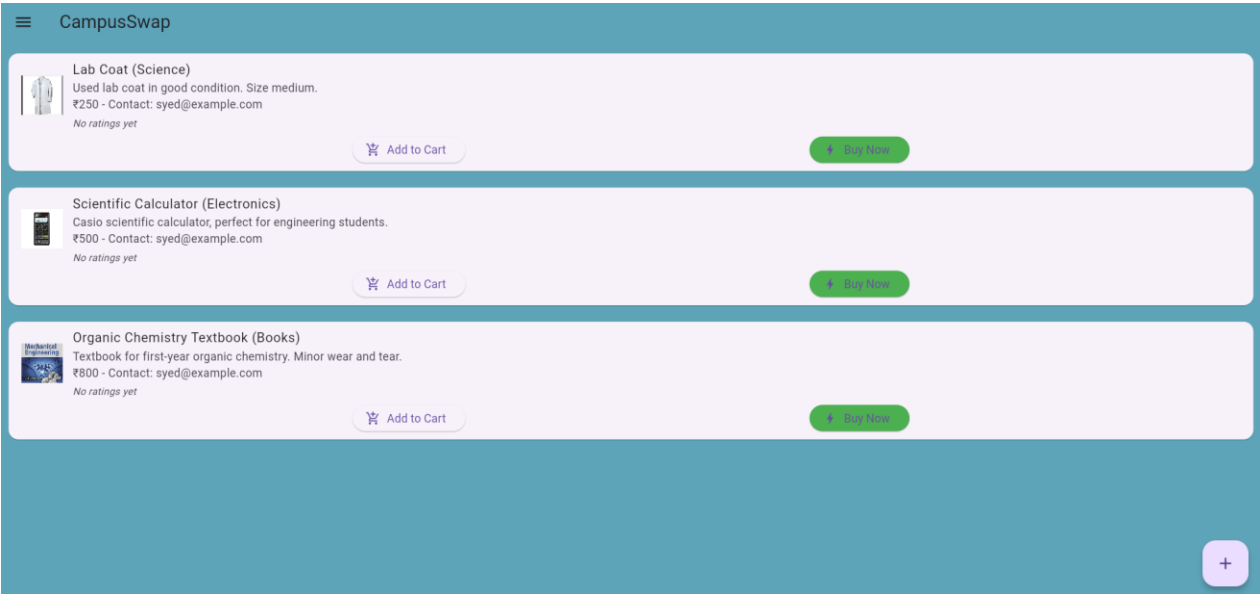


The "Create Account" screen has a teal background. At the top, there is a white header bar with a back arrow and the text "Create Account". Below this, there are several input fields: "Name", "Date of Birth" (with a calendar icon on the right), "USN", "Email", "Branch" (with a dropdown arrow on the right), "Year of Study" (with a dropdown arrow on the right), and "Password". At the bottom of the screen is a white button labeled "Create Account".

THEN THE USER NEED TO LOGIN INTO THE PAGE

Home Screen

After logging in, the user is presented with a list of items available for sale. Each item is displayed as a card with an image, title, price, and buttons to add to the cart or buy directly.

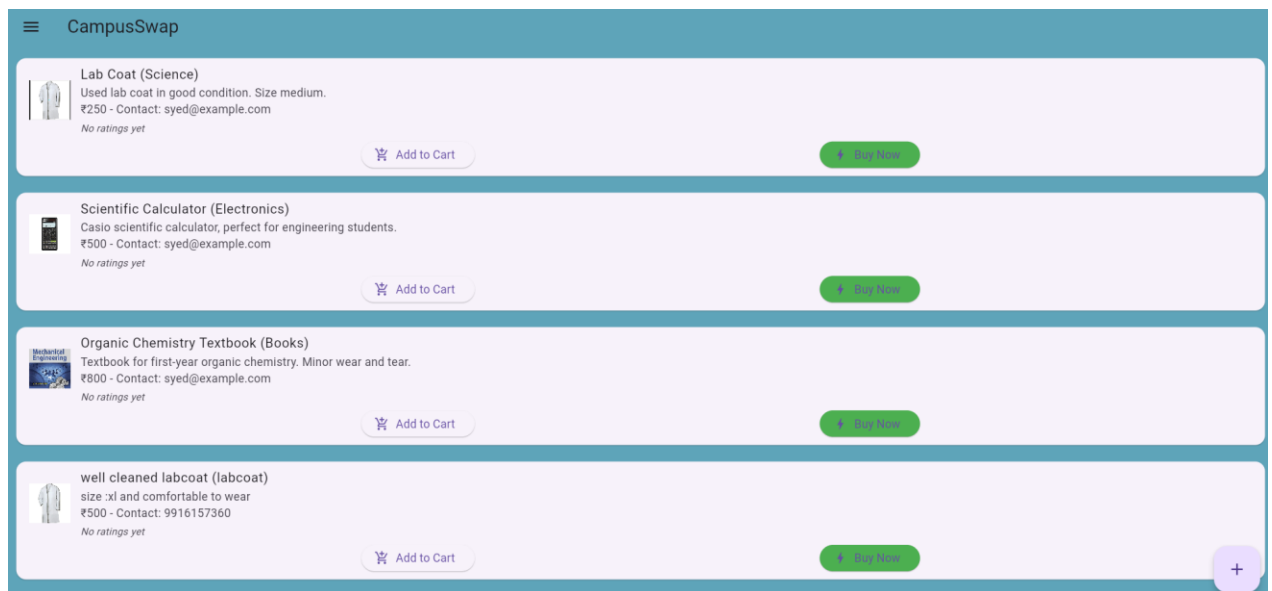


Add Item Screen

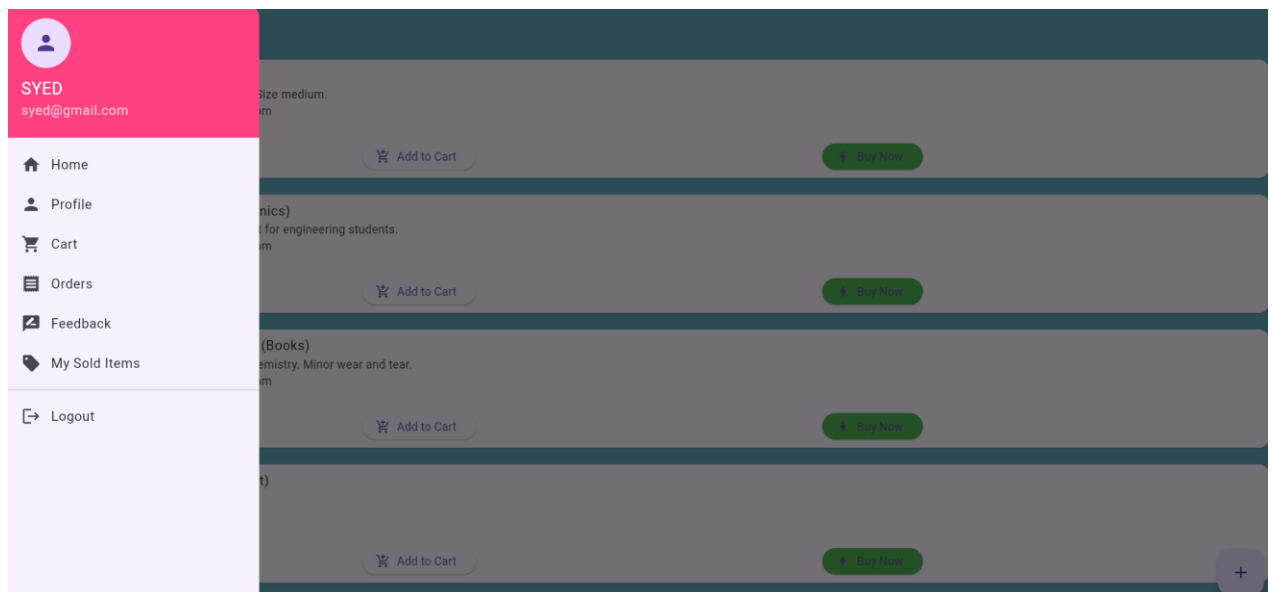
This screen allows sellers to add a new product. It is a form with fields for product details and a button to upload images.

The screenshot shows the 'Add New Item' form screen. It has a light purple header with a back arrow and the text 'Add New Item'. The form itself is on a darker purple background and contains several input fields: 'Category', 'Title', 'Description', 'Price', 'Contact', and 'UPI ID'. Below these fields is a white bar with a camera icon and the text 'Pick Images'. Underneath that, it says 'No images selected'. At the bottom of the form is a white bar with the text 'Submit Item'.

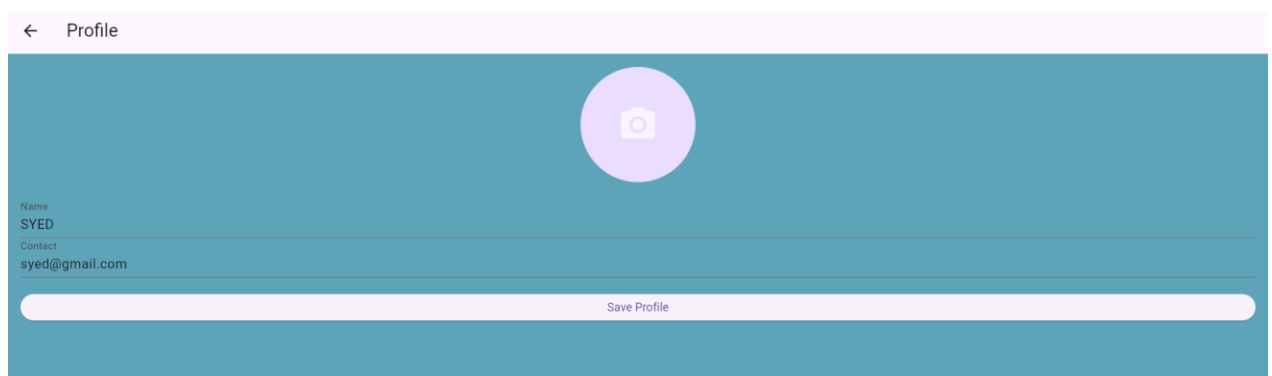
After adding one item



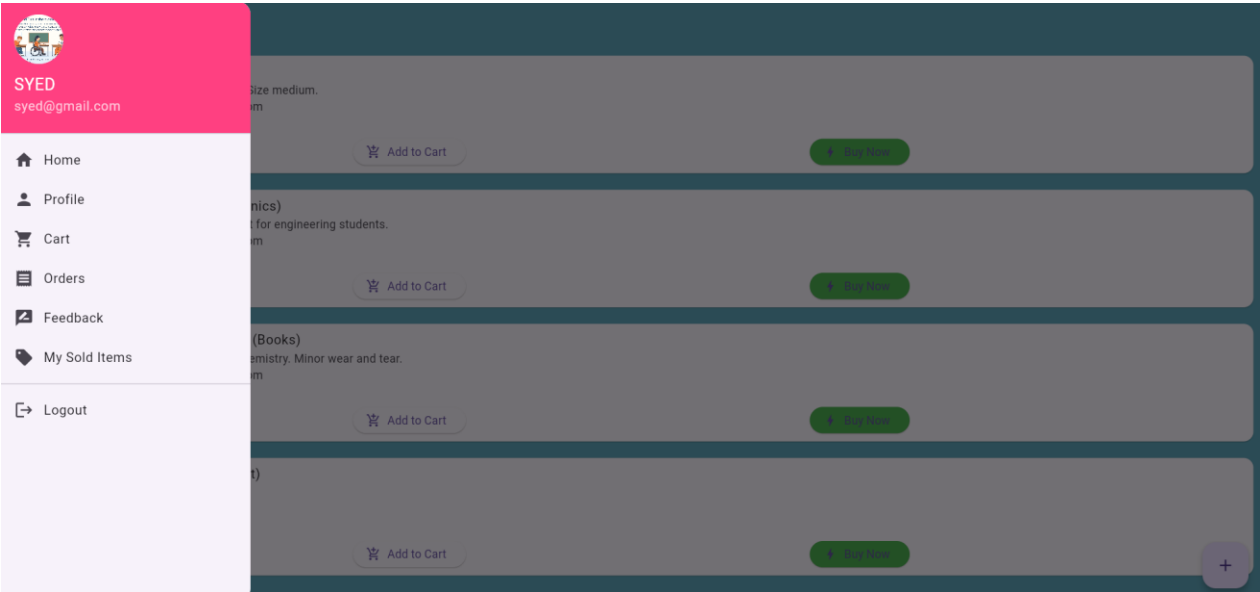
Draw menu : where user can find several details or buttons



Profile update :- The user can update his/her profile



After updation :-



Add to cart screen :- The items added in cart can be viewed in cart section and the user can place the order



← Buy Now

Order Summary

Lab Coat
₹250

Total: ₹250

Payment Method

☐ UPI

Pay to seller's UPI when you meet (UPI ID is provided by seller).

☐ Cash

Pay in cash on delivery/pickup.

Your Contact

Preferred Place of Delivery


✓ Confirm Purchase

Order screen :- when user place the order he/she can see the ordered item in order section

← Your Orders

Order Placed: 2025-08-12
Buyer Contact: 9916157360
Delivery Place: hostel

Items:

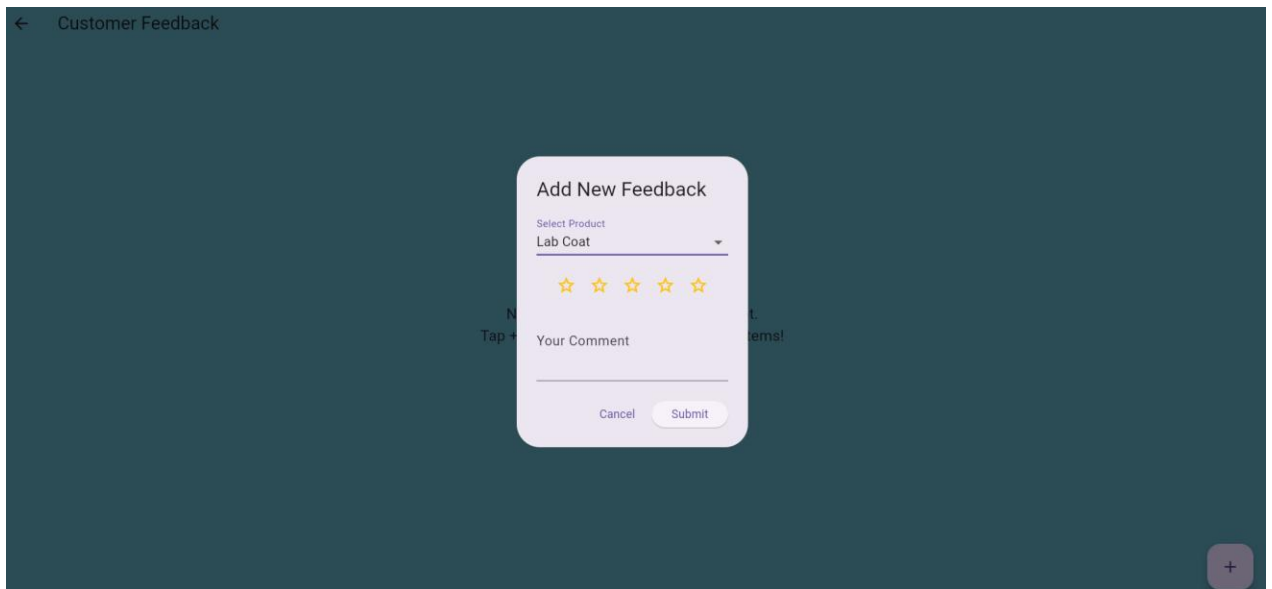


Lab Coat - ₹250
Category: Science

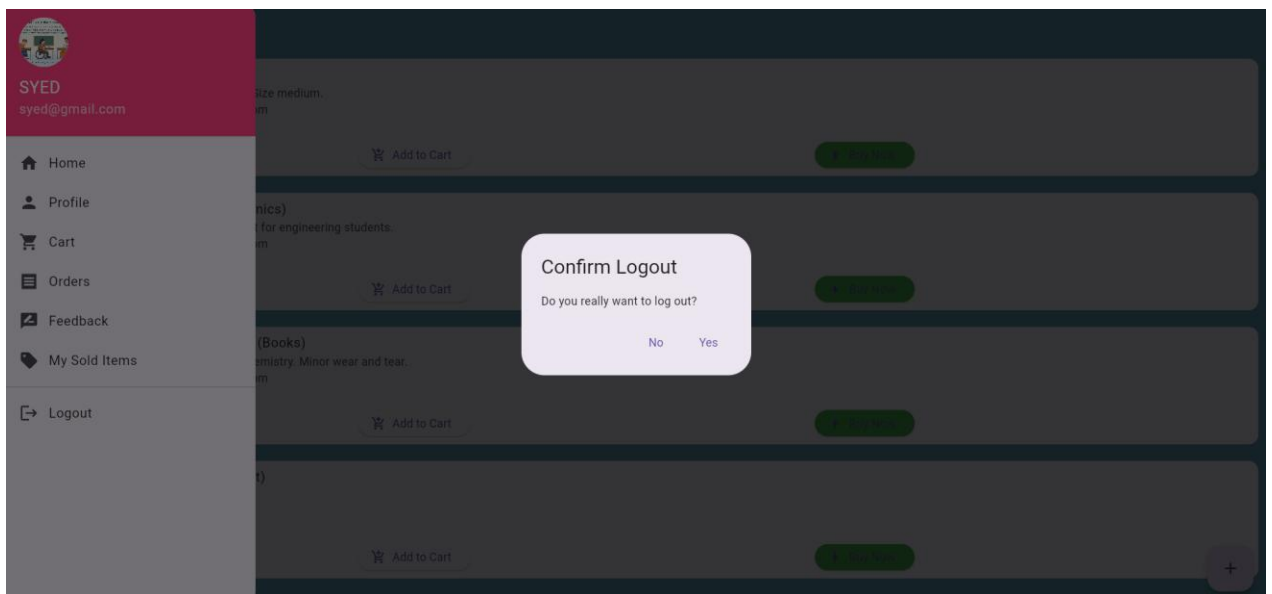
Total: ₹250

📦 Confirm Delivery

Feedback screen :- When user gets the delivery he can give feed back on items received



Logout screen:- when user click on logout it pop up with alert message where user is asked to confirm logout



4. Conclusion

The CampusSwap project stands as a compelling and well-executed proof-of-concept for a student-centric mobile marketplace. Its core strength lies in its **modular, clean architecture** built on Flutter, demonstrating how a complex application can be both functional and intuitive. The app successfully simulates a complete e-commerce ecosystem, from user authentication and item listings to a functional shopping cart and order management system. This not only validates the app's core idea but also provides a **robust, scalable foundation** for future enhancements.

In conclusion, CampusSwap is more than just a prototype; it's a testament to the power of targeted, community-driven platforms. It's ready to transition from a successful project into a vibrant, real-world application that can transform the campus experience by fostering sustainability and connection among students.