



CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF TECHNOLOGY AND ENGINEERING
CHANDUBHAI S. PATEL INSTITUTE OF TECHNOLOGY
DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING



Subject : Mobile Application Development

Semester: 5

Subject Code: AIML308

Academic Year :2025-26(ODD)

NAME : CHOKSI OM CHIRAGBHAI

ID: 23AIML010

Practical 7

Problem Definition

You are building a mobile application where users navigate through multiple screens like login, dashboard, and profile. Design a Product Catalog App using GridView and custom cards with images.

Supplementary Problems -	Gallery app or Recipe app
--------------------------	---------------------------

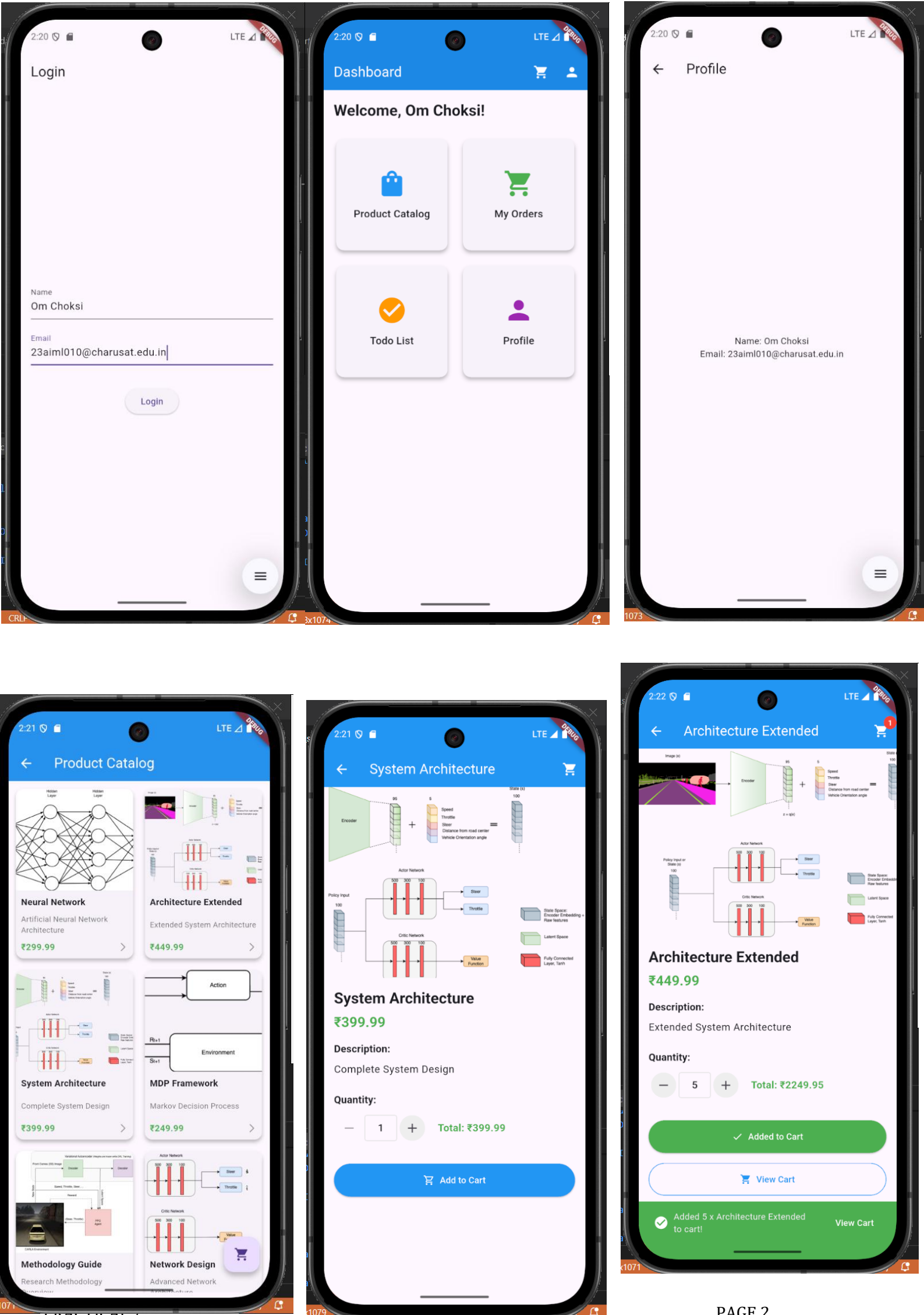
Technical Approach:

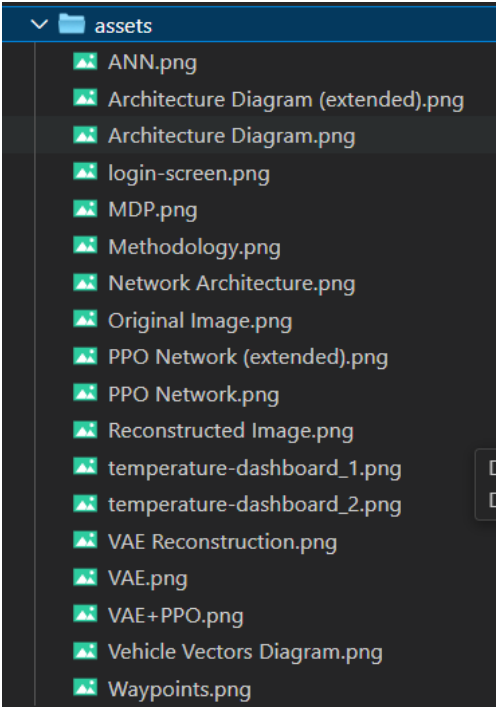
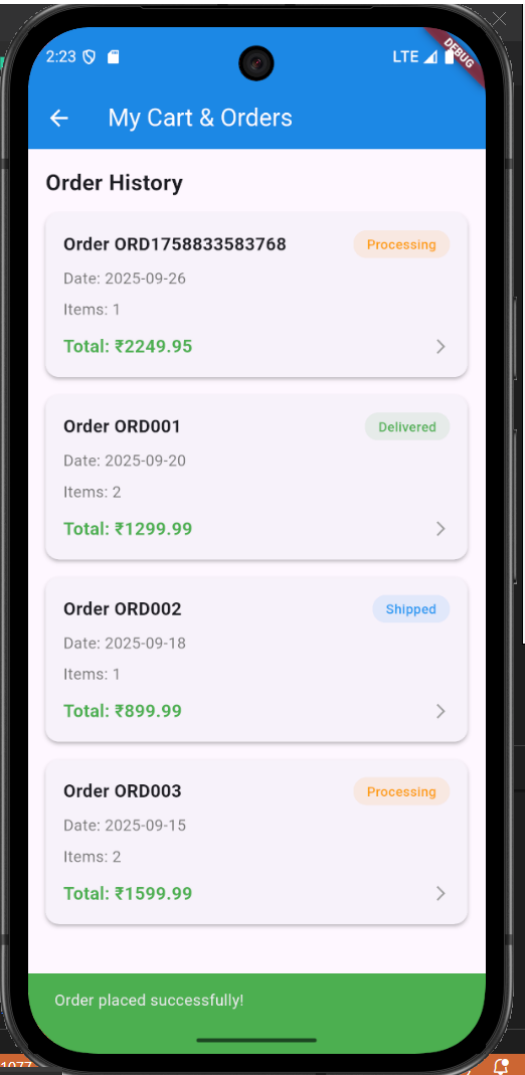
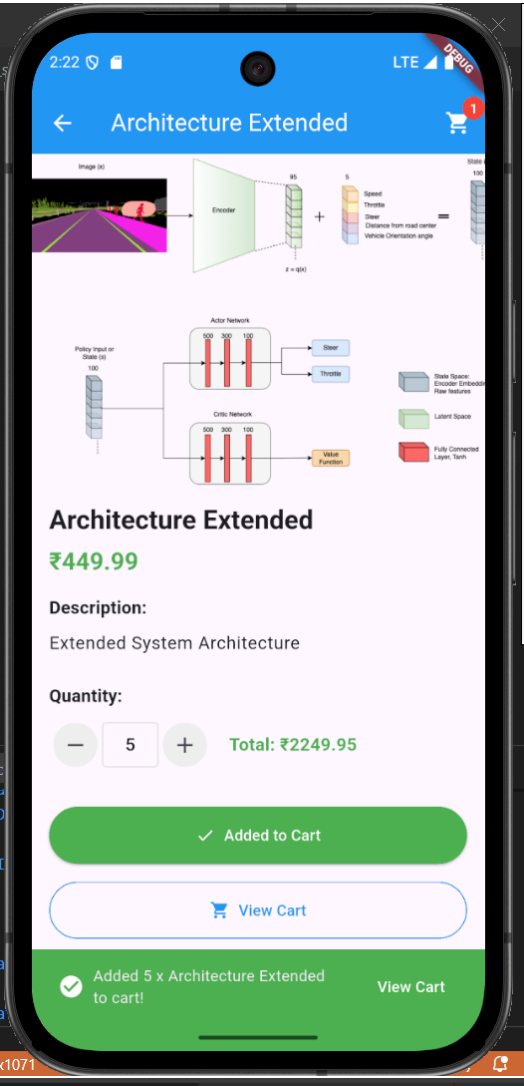
The application implements a product catalog using Flutter's GridView for responsive grid layouts, with custom reusable ProductCard widgets ensuring consistent UI design. State management is achieved through the Provider package, utilizing a CartManager class that extends ChangeNotifier for dynamic cart functionality across screens. Core widgets include StatefulWidget for interactive product details and cart management, GridView for product display, and Image.asset for loading local assets, with setState handling local UI updates in quantity selectors and form validations.

File Structure:

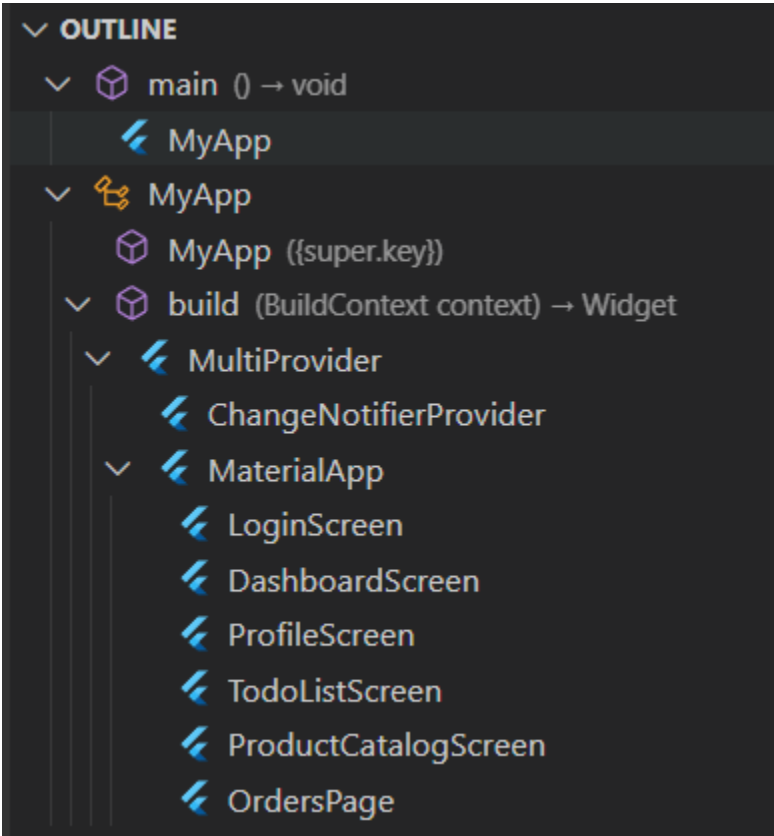
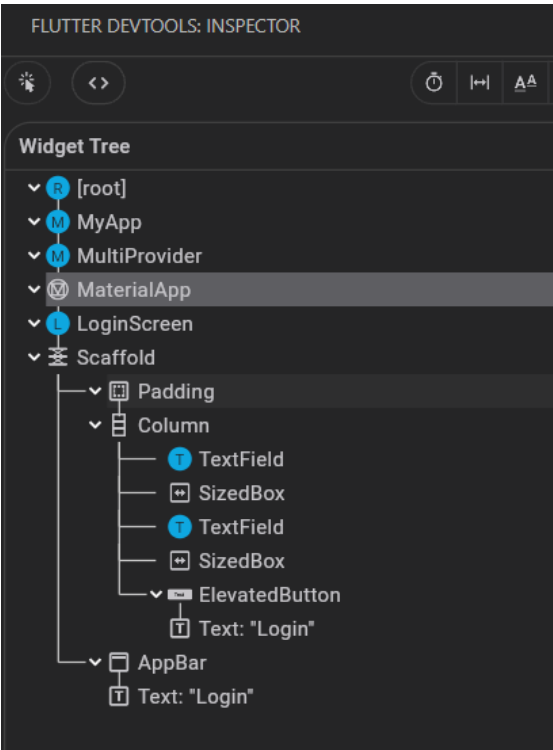
```
lib/  
├── cart_manager.dart // Cart management: add/remove items, update quantities.  
├── dashboard.dart // Main dashboard: app overview and navigation.  
├── login.dart // User auth: login and registration.  
├── main.dart // App entry point: initializes root widget.  
├── orders.dart // Order management: history, details, placement.  
├── product_catalog.dart // Product catalog: browse and search items.  
├── product_detail.dart // Product details: info, images, options.  
├── profile.dart // User profile: details, settings, account.  
├── todolist.dart // To-do list: create, edit, track tasks.  
└── widgets/  
    └── product_card.dart // Reusable product card widget for lists/catalogs.
```

Screenshots:





```
pubspec.yaml X product_detail.dart product_c
pubspec.yaml
54 flutter:
60
61   assets:
62     - assets/
63
64   # To add assets to your application, add
65   # assets:
66   #   - images/a_dot_burr.jpeg
67   #   - images/a_dot_ham.jpeg
68
69   # An image asset can refer to one or more
```



Key Questions:**1. How to create reusable custom widgets?**

Ans: Create a separate StatelessWidget or StatefulWidget class in a dedicated file (e.g., lib/widgets/product_card.dart), define required parameters in the constructor, and use it across multiple screens by importing and instantiating the widget with appropriate props, as demonstrated by the ProductCard widget used in the GridView.

2. How to use GridView for layout?

Ans: Use GridView.builder or GridView.count with SliverGridDelegateWithFixedCrossAxisCount to specify cross-axis count, spacing, and aspect ratio, then provide an itemBuilder function that returns widgets for each grid item, as implemented in the ProductCatalogScreen for displaying product cards in a 2-column grid.

3. How to load local assets (images)?

Ans: Add image files to the assets/ directory, configure the path in pubspec.yaml under the assets section, then use Image.asset() widget with the asset path string, including error handling with errorBuilder to display fallback content if the image fails to load, as shown in the ProductCard widget.

Key Skills to be Understand:

UI design, Reusability, Grid layout

Design reusable and responsive UI