Bharatiya Vidya Bhavan's
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri (W), Mumbai – 400 058.
Department of Computer Science and Engineering

Experiment	3
Aim	Create an E-Commerce app using Bottom Navigation
Objective	● To Create E-Commerce App ● To Develop Bottom Navigation ● To Use 60-30-10 rule
Name	Vedant Ekanath Onkar
UCID	2024510036
Class	FY MCA
Batch	B
Date of Submission	11-Mar-2025

Technology used	
Task	Develop an e-commerce app with Bottom Navigation Bar with 4 tabs: Home, Browse list of Products (Atleast 5 products) and see product details (image, name, price, and description), Profile and Cart.

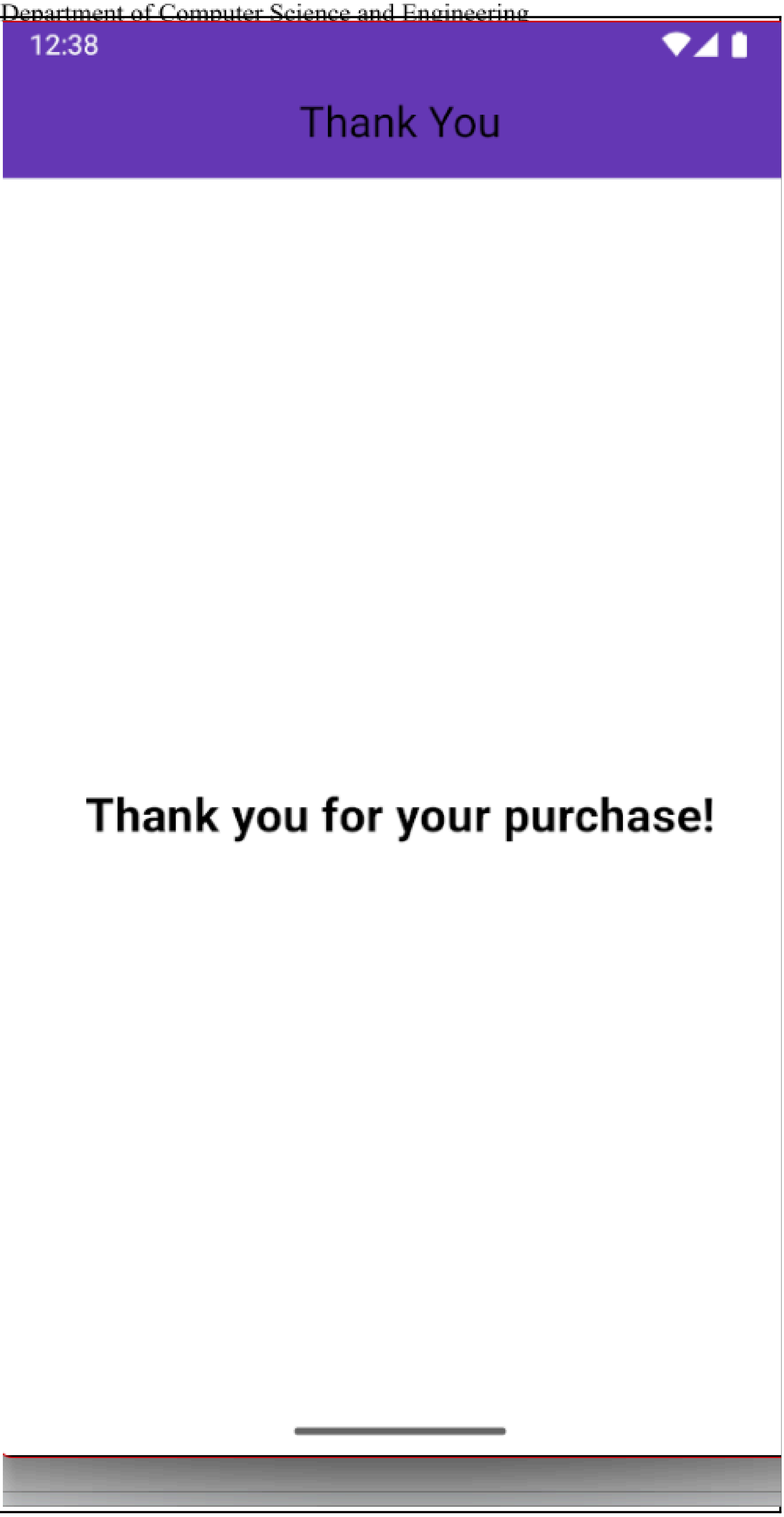
Code with proper label

```
import 'package:flutter/material.dart';

void main() => runApp(const MyApp());

List<CartItem> cartItems = [];

double parsePrice(dynamic value) {
  try {
    return double.parse(value.toString());
  } catch (e) {
    print("Error parsing price: $e");
    return 0.0;
  }
}

class CartItem {
  final Map<String, dynamic> product;
  int quantity;
  CartItem({required this.product, this.quantity = 1});
  double get totalPrice => parsePrice(product['price']) *
quantity;
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'LowKeyboards',
      theme: ThemeData(
        primarySwatch: Colors.deepPurple,
        colorScheme: ColorScheme.fromSwatch(primarySwatch:
Colors.deepPurple)
            .copyWith(secondary: Colors.amber),
        scaffoldBackgroundColor: Colors.white,
        appBarTheme: const AppBarTheme(
          backgroundColor: Colors.deepPurple,
          centerTitle: true,
        ),
        bottomNavigationBarTheme: const
BottomNavigationBarThemeData(
          selectedItemColor: Colors.amber,
          unselectedItemColor: Colors.white70,
          backgroundColor: Colors.deepPurple,
        ),
      ),
      home: const BottomNavController(),
    );
  }
}

class BottomNavController extends StatefulWidget {
  const BottomNavController({Key? key}) : super(key: key);
  @override
  _BottomNavControllerState createState() =>
_BottomNavControllerState();
}

class _BottomNavControllerState extends
State<BottomNavController> {
  int _selectedIndex = 0;
  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }
  @override
  Widget build(BuildContext context) {
    final List<Widget> _pages = [
      HomePage(onExplore: () => _onItemTapped(1)),
      DashboardPage(),
      CartPage(),
    ];
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'LowKeyboards',
          style: TextStyle(
            fontFamily: 'Montserrat',
            fontWeight: FontWeight.bold,
            fontSize: 24,
            color: Colors.white,
          ),
        ),
        centerTitle: true,
        actions: [
          TextButton(
            onPressed: () {
              _onItemTapped(0);
              print("Navigated to Home");
            },
            child: Text(
              'Home',
              style: TextStyle(
                fontFamily: 'Montserrat',
                fontSize: 16,
                color: _selectedIndex == 0 ? Colors.amber :
Colors.white70,
              ),
            ),
          ),
          TextButton(
            onPressed: () {
              _onItemTapped(1);
              print("Navigated to Products");
            },
            child: Text(
              'Products',
              style: TextStyle(
                fontFamily: 'Montserrat',
                fontSize: 16,
                color: _selectedIndex == 1 ? Colors.amber :
Colors.white70,
              ),
            ),
          ),
          TextButton(
            onPressed: () {
              _onItemTapped(2);
              print("Navigated to Cart");
            },
            child: Text(
              'Cart',
              style: TextStyle(
                fontFamily: 'Montserrat',
                fontSize: 16,
                color: _selectedIndex == 2 ? Colors.amber :
Colors.white70,
              ),
            ),
          ),
        ],
      ),
      body: _pages[_selectedIndex],
      bottomNavigationBar: BottomNavigationBar(
        currentIndex: _selectedIndex,
        onTap: (int index) {
          setState(() {
            _selectedIndex = index;
          });
        },
        items: const [
          BottomNavigationBarItem(icon: Icon(Icons.home),
label: 'Home'),
          BottomNavigationBarItem(icon:
Icon(Icons.dashboard), label: 'Products'),
          BottomNavigationBarItem(icon:
Icon(Icons.shopping_cart), label: 'Cart'),
        ],
      ),
    );
  }
}

class HomePage extends StatelessWidget {
  final VoidCallback onExplore;
  const HomePage({Key? key, required this.onExplore}) :
super(key: key);
  final List<Map<String, dynamic>> featuredKeyboards = const
[
    {
      'name': 'LowKey Pro',
      'image': 'lib/images/reddragon-k616.png',
      'description': 'A premium mechanical keyboard with
```

```
vibrant RGB lighting.',
      'price': 99.99,
    },
    {
      'name': 'Wireless Xpress',
      'image': 'lib/images/razer.png',
      'description': 'A sleek wireless keyboard built for
speed and efficiency.',
      'price': 79.99,
    },
  ];
  @override
  Widget build(BuildContext context) {
    const double centerBoxHeight = 170;
    return SafeArea(
      child: Column(
        children: [
          Expanded(
            flex: 1,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: const [
                Text(
                  "Welcome to LowKeyboards",
                  style: TextStyle(
                    fontFamily: 'Montserrat',
                    fontSize: 22,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                SizedBox(height: 10),
                Row(
                  mainAxisAlignment:
MainAxisAlignment.center,
                  children: [
                    Icon(Icons.star, color: Colors.amber),
                    SizedBox(width: 10),
                    Icon(Icons.keyboard, color:
Colors.deepPurple),
                    SizedBox(width: 10),
                    Icon(Icons.lightbulb, color:
Colors.amber),
                  ],
                ),
              ],
            ),
          ),
          Expanded(
            flex: 2,
            child: Container(
              margin: const EdgeInsets.symmetric(horizontal:
20),
              padding: const EdgeInsets.all(12),
              decoration: BoxDecoration(
                color: Colors.white,
                borderRadius: BorderRadius.circular(16),
                border: Border.all(color: Colors.deepPurple,
width: 2),
              ),
              child: PageView.builder(
                controller: PageController(viewportFraction:
0.9),
                itemCount: featuredKeyboards.length,
                itemBuilder: (context, index) {
                  var product = featuredKeyboards[index];
                  double price =
parsePrice(product['price']);
                  return Container(
                    margin: const
EdgeInsets.symmetric(horizontal: 8),
                    child: Material(
                      elevation: 3,
                      borderRadius:
BorderRadius.circular(12),
                      child: InkWell(
                        borderRadius:
BorderRadius.circular(12),
                        onTap: () {
                          print("Tapped on
${product['name']}");
                          Map<String, dynamic> productCopy =
Map.from(product);
                          Navigator.push(
                            context,
                            MaterialPageRoute(
                              builder: (_) =>
ProductDetailPage(product: productCopy),
                            ),
                          );
                        },
                        child: Column(
                          crossAxisAlignment:
```

```
CrossAxisAlignment.stretch,
                          children: [
                            SizedBox(
                              height: centerBoxHeight *
0.75,
                              child: ClipRRect(
                                borderRadius: const
BorderRadius.vertical(top: Radius.circular(12)),
                                child: Image.asset(
                                  product['image'],
                                  fit: BoxFit.cover,
                                ),
                              ),
                            ),
                            Padding(
                              padding: const
EdgeInsets.all(4.0),
                              child: Text(
                                product['name'],
                                style: const TextStyle(
                                  fontSize: 14,
                                  fontWeight:
FontWeight.bold,
                                  fontFamily: 'Montserrat',
                                ),
                                textAlign: TextAlign.center,
                              ),
                            ),
                            Padding(
                              padding: const
EdgeInsets.symmetric(horizontal: 4.0),
                              child: Text(

"\$${price.toStringAsFixed(2)}",
                                style: const
TextStyle(fontSize: 12),
                                textAlign: TextAlign.center,
                              ),
                            ),
                          ],
                        ),
                      ),
                    ),
                  );
                },
              ),
            ),
          ),
          Expanded(
            flex: 1,
            child: Center(
              child: ElevatedButton(
                onPressed: onExplore,
                child: const Text(
                  "Explore More Products",
                  style: TextStyle(fontFamily: 'Montserrat', fontSize: 16),
                ),
              ),
            ),
          ),
        ],
      ),
    );
  }
}

class DashboardPage extends StatelessWidget {
  final List<Map<String, dynamic>> allKeyboards = const [
    {
      'name': 'LowKey Pro',
      'image': 'lib/images/apex.jpg',
      'description': 'A premium mechanical keyboard with vibrant RGB lighting.',
      'price': 99.99,
    },
    {
      'name': 'Wireless Xpress',
      'image': 'lib/images/razer.png',
      'description': 'A sleek wireless keyboard built for speed and efficiency.',
      'price': 79.99,
    },
    {
      'name': 'Gamer\'s Edge',
      'image': 'lib/images/rdg-deimos.png',
      'description': 'Optimized for gaming with programmable keys and ergonomic design.',
      'price': 129.99,
    },
    {
      'name': 'Portable Swift',
      'image': 'lib/images/reddragon-k616.png',
      'description': 'Compact and lightweight, perfect for
productivity on the go.',
      'price': 59.99,
    },
    {
      'name': 'Ergo Comfort',
      'image': 'lib/images/rog.png',
      'description': 'Designed for long typing sessions with
ergonomic features.',
      'price': 109.99,
    },
  ];
  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: ListView.builder(
        padding: const EdgeInsets.symmetric(horizontal: 20,
vertical: 20),
        itemCount: allKeyboards.length,
        itemBuilder: (context, index) {
          var product = allKeyboards[index];
          double price = parsePrice(product['price']);
          return Container(
            margin: const EdgeInsets.only(bottom: 20),
            child: Material(
              elevation: 3,
              borderRadius: BorderRadius.circular(16),
              child: InkWell(
                borderRadius: BorderRadius.circular(16),
                onTap: () {
                  print("Tapped on ${product['name']} in
Dashboard");
                  Map<String, dynamic> productCopy =
Map.from(product);
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (_) =>
ProductDetailPage(product: productCopy),
                    ),
                  );
                },
                child: Padding(
                  padding: const EdgeInsets.all(12),
                  child: Column(
                    crossAxisAlignment:
CrossAxisAlignment.stretch,
                    children: [
                      AspectRatio(
                        aspectRatio: 16 / 9,
                        child: ClipRRect(
                          borderRadius:
BorderRadius.circular(12),
                          child: Image.asset(
                            product['image'],
                            fit: BoxFit.cover,
                          ),
                        ),
                      ),
                      const SizedBox(height: 10),
                      Text(
                        product['name'],
                        style: const TextStyle(
                          fontFamily: 'Montserrat',
                          fontSize: 18,
                          fontWeight: FontWeight.bold,
                        ),
                        textAlign: TextAlign.center,
                      ),
                      const SizedBox(height: 5),
                      Text(
                        "\$${price.toStringAsFixed(2)}",
                        style: const TextStyle(
                          fontFamily: 'Montserrat',
                          fontSize: 16,
                        ),
                        textAlign: TextAlign.center,
                      ),
                    ],
                  ),
                ),
              ),
            ),
          );
        },
      ),
    );
  }
}

class CartPage extends StatefulWidget {
  @override
  _CartPageState createState() => _CartPageState();
}

class _CartPageState extends State<CartPage> {
  double get totalSum =>
      cartItems.fold(0.0, (sum, item) => sum +
item.totalPrice);
  void _navigateToBilling() {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (_) => BillingPage(total:
totalSum)),
    );
  }
  void _clearCart() {
    setState(() {
      cartItems.clear();
      print("Cart cleared");
    });
  }
  void _reduceQuantity(int index) {
    setState(() {
      cartItems[index].quantity--;
      if (cartItems[index].quantity <= 0) {
        cartItems.removeAt(index);
      }
    });
  }
  void _increaseQuantity(int index) {
    setState(() {
      cartItems[index].quantity++;
    });
  }
  @override
  Widget build(BuildContext context) {
    print("Building CartPage with ${cartItems.length}
items");
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: cartItems.isEmpty
          ? const Center(child: Text('Your cart is empty',
style: TextStyle(fontSize: 18)))
          : Column(
              crossAxisAlignment:
CrossAxisAlignment.stretch,
              children: [
                Expanded(
                  child: ListView.builder(
                    itemCount: cartItems.length,
                    itemBuilder: (context, index) {
                      final item = cartItems[index];
                      double itemPrice =
parsePrice(item.product['price']);
                      return ListTile(
                        contentPadding: const
EdgeInsets.symmetric(horizontal: 8.0),
                        leading: Image.asset(
                          item.product['image'],
                          width: 50,
                          height: 50,
                          fit: BoxFit.cover,
                        ),
                        title: Text(item.product['name']),
                        subtitle: Row(
                          children: [
                            IconButton(
                              icon: const
Icon(Icons.remove),
                              onPressed: () =>
_reduceQuantity(index),
                            ),
                            Text('Qty: ${item.quantity}'),
                            IconButton(
                              icon: const Icon(Icons.add),
                              onPressed: () =>
_increaseQuantity(index),
                            ),
                            const SizedBox(width: 8),
                            Text('Price:
\$${itemPrice.toStringAsFixed(2)}'),
                          ],
                        ),
                        trailing:
Text('\$${item.totalPrice.toStringAsFixed(2)}'),
                      );
                    },
                  ),
                ),
                Text(
                  'Total: \$${totalSum.toStringAsFixed(2)}',
                  style: const TextStyle(fontSize: 20,
fontWeight: FontWeight.bold),
                  textAlign: TextAlign.right,
                ),
                const SizedBox(height: 10),
                ElevatedButton(
                  onPressed: _navigateToBilling,
                  child: const Text('Checkout'),
                ),
                const SizedBox(height: 10),
                ElevatedButton(
                  onPressed: _clearCart,
                  child: const Text('Clear Cart'),
                  style:
ElevatedButton.styleFrom(backgroundColor: Colors.redAccent),
                ),
              ],
            ),
    );
  }
}

class BillingPage extends StatelessWidget {
  final double total;
  const BillingPage({Key? key, required this.total}) :
super(key: key);
  void _pay(BuildContext context) {
    cartItems.clear();
    Navigator.pushAndRemoveUntil(
      context,
      MaterialPageRoute(builder: (_) => const ThanksPage()),
      (route) => false,
    );
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Billing')),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: [
            Text('Total Amount:
\$${total.toStringAsFixed(2)}',
                style: const TextStyle(fontSize: 24,
fontWeight: FontWeight.bold)),
            const SizedBox(height: 20),
            ElevatedButton(
              onPressed: () => _pay(context),
              child: const Text('Pay'),
            ),
          ],
        ),
      ),
    );
  }
}

class ThanksPage extends StatelessWidget {
  const ThanksPage({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Thank You')),
      body: const Center(
        child: Text('Thank you for your purchase!', style:
TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),
      ),
    );
  }
}

class ProductDetailPage extends StatelessWidget {
  final Map<String, dynamic> product;
  const ProductDetailPage({Key? key, required this.product})
: super(key: key);
  void _addToCart(BuildContext context) {
    print("Adding to cart: ${product['name']}, Price:
${product['price']}");
    final index = cartItems.indexWhere((item) =>
item.product['name'] == product['name']);
    if (index >= 0) {
      cartItems[index].quantity++;
      print("Increased quantity for ${product['name']} to
${cartItems[index].quantity}");
    } else {
      Map<String, dynamic> productCopy = Map.from(product);
      cartItems.add(CartItem(product: productCopy));
      print("Added new item to cart: ${product['name']}");
    }
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text('${product['name']} added to
cart')),
    );
  }
  @override
  Widget build(BuildContext context) {
    double price = parsePrice(product['price']);
    return DefaultTabController(
      length: 3,
      child: Scaffold(
        appBar: AppBar(
          title: Text(product['name']),
          bottom: const TabBar(
            indicatorColor: Colors.amber,
            tabs: [
              Tab(text: "Information"),
              Tab(text: "Specification"),
              Tab(text: "Dealer"),
            ],
          ),
        ),
        body: SingleChildScrollView(
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
              AspectRatio(
                aspectRatio: 16 / 9,
                child: Image.asset(
                  product['image'] ??
'lib/images/placeholder.png',
                  fit: BoxFit.cover,
                ),
              ),
              const SizedBox(height: 20),
              Padding(
                padding: const
EdgeInsets.symmetric(horizontal: 16.0),
                child: Text(
                  product['name'],
                  style: const TextStyle(fontSize: 24,
fontWeight: FontWeight.bold),
                  textAlign: TextAlign.center,
                ),
              ),
              const SizedBox(height: 10),
              Padding(
                padding: const
EdgeInsets.symmetric(horizontal: 16.0),
                child: Text(
                  product['description'],
                  style: const TextStyle(fontSize: 16),
                  textAlign: TextAlign.center,
                ),
              ),
              const SizedBox(height: 10),
              Padding(
                padding: const
EdgeInsets.symmetric(horizontal: 16.0),
                child: Text(
                  'Price: \$${price.toStringAsFixed(2)}',
                  style: const TextStyle(fontSize: 18,
fontWeight: FontWeight.bold),
                  textAlign: TextAlign.center,
                ),
              ),
              const SizedBox(height: 20),
              Container(
                height: 300,
                child: const TabBarView(
                  children: [
                    Padding(
                      padding: EdgeInsets.all(16.0),
                      child: Text(
                        "Our LowKey Pro is engineered for
both performance and style. It features precision mechanical
switches, customizable RGB lighting, and a durable build for
long hours of use.",
                        style: TextStyle(fontSize: 16),
                      ),
                    ),
                    Padding(
                      padding: EdgeInsets.all(16.0),
                      child: Text(
                        "Specifications:\n• Switch Type:
Cherry MX Red\n• Key Rollover: N-Key\n• Connectivity: Wired
USB\n• Dimensions: 45 x 15 x 4 cm",
                        style: TextStyle(fontSize: 16),
                      ),
                    ),
                    Padding(
                      padding: EdgeInsets.all(16.0),
                      child: Text(
                        "Dealer Information:\nDealer:
TechGear Inc.\nPhone: (555) 123-4567\nEmail:
```

```
support@techgear.com",
                        style: TextStyle(fontSize: 16),
                      ),
                    ),
                  ],
                ),
              ),
              const SizedBox(height: 20),
              Padding(
                padding: const
EdgeInsets.symmetric(horizontal: 16.0),
                child: ElevatedButton(
                  onPressed: () => _addToCart(context),
                  child: const Text('Add to Cart'),
                ),
              ),
              const SizedBox(height: 20),
            ],
          ),
        ),
      ),
    );
  }
}
```

Screenshots	 Thank you for your purchase!

Question and Answers	1. How to create Bottom Navigation? We used the BottomNavigationBar widget inside a Scaffold, managing the selected tab with a StatefulWidget that updates the page based on the tapped index. 2. How did you use the 60-30-10 rule in your application? We applied deep purple as the dominant color (60%), white for backgrounds (30%), and amber for accents like active navigation items (10%) to achieve a balanced look. 3. Which new elements did you use for creating UI components? We used PageView.builder for a snapping horizontal scroll, wrapped product cards in Material and InkWell for interactive feedback, and DefaultTabController with TabBar for tabbed content in the product detail page.
Conclusion	In this project, we learned how to build a complete Flutter app with a bottom navigation bar to manage different pages. We applied the 60-30-10 rule for a balanced color scheme and used advanced UI elements like PageView for snapping horizontal scrolling, Material with InkWell for interactive product cards, and DefaultTabController with TabBar for tabbed content in the product details page. Overall, we deepened our understanding of responsive layout design, state management, and navigation in Flutter.