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	9&10
Aim	Capstone project with database connectivity - Sneakers E-commerce (Sneaker Store) with firebase connectivity
Objective	Develop a modern and visually appealing e-commerce app for sneakers using Flutter. Ensure cross-platform compatibility (Android, iOS, Web) with responsive UI. Implement Firebase Authentication for secure user login/signup. Use Cloud Firestore to manage sneaker product listings, categories, and user data in real-time.
Name	Panchangni Dhangar Shreyas Gaikwad Sanjana Palkar
UCID	2024510014 2024510019 2024510040
Class	FYMCA
Batch	D
Date of Submission	5/5/2025

Technology used	**Flutter, Vs Code, Firebase**
Task	Set up Flutter project and integrate Firebase (Auth, Firestore, Storage, FCM). Implement user login/signup with Firebase Authentication. Design UI for product listing, details, cart, and checkout. Fetch sneaker data from Firestore and display it.

Code with proper label	main.dart
	
	import 'package:firebase_core/firebase_core.dart';
	import 'package:flutter/material.dart';
	import 'package:flutter_screenutil/flutter_screenutil.dart';
	import 'package:provider/provider.dart';
	import 'package:hive_flutter/hive_flutter.dart';
	import 'package:sneaker_app/controllers/mainscreen_provider.dart';
	import 'package:sneaker_app/controllers/product_provider.dart';
	import 'package:sneaker_app/views/pages/cart_page.dart';
	import 'package:sneaker_app/views/pages/favorites.dart';
	import 'package:sneaker_app/views/pages/login_signup_page.dart';
	import 'package:sneaker_app/views/pages/splash_screen.dart';
	import 'firebase_options.dart';
	import 'package:flutter/foundation.dart' show kIsWeb;
	
	// PAGES
	import '../views/pages/mainscreen.dart';
	import 'controllers/cart_provider.dart';
	import 'controllers/favorites_provider.dart';
	
	void main() async {
	WidgetsFlutterBinding.ensureInitialized();
	try {

```
  if (kIsWeb) {
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
 } else {
  await Firebase.initializeApp();
 }
  await Hive.initFlutter();
 await Hive.openBox("cart_box");
 await Hive.openBox("fav_box");
 runApp(
  MultiProvider(
   providers: [
    ChangeNotifierProvider(create: (context) => MainScreenNotifier()),
    ChangeNotifierProvider(create: (context) => ProductNotifier()),
    ChangeNotifierProvider(create: (context) => FavouritesNotifier()),
    ChangeNotifierProvider(create: (context) => CartProvider())
   ],
   child: const MyApp(),
  ),
 );
 } catch (e, stacktrace) {
  print('Firebase initialization error: $e');
  print('Stacktrace: $stacktrace');
 }
}

class MyApp extends StatelessWidget {
 const MyApp({super.key});
 @override
 Widget build(BuildContext context) {
  return ScreenUtilInit(
   designSize: const Size(375, 812),
   minTextAdapt: true,
   splitScreenMode: true,
   builder: (context, child) {
    return MaterialApp(
     debugShowCheckedModeBanner: false,
     routes: {
      '/main': (context) => const MainScreen(userEmail: 'example@example.com'),
      '/favorites': (context) => const Favourites(),
      '/cart': (context) => const CartPage(),
      '/login': (context) => const LoginSignupScreen(),
      '/splash': (context) => const SplashScreen(),
     },
     initialRoute: '/splash',
    );
   },
  );
 }
}
```

cart_page.dart

```
import 'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:flutter_slidable/flutter_slidable.dart';
import 'package:flutter_vector_icons/flutter_vector_icons.dart';
import 'package:hive/hive.dart';
```

```
import 'package:provider/provider.dart';
import 'package:sneaker_app/views/pages/checkout_page.dart';
import 'package:sneaker_app/views/shared/app_style.dart';
import '../../controllers/cart_provider.dart';
import '../shared/checkout_btn.dart';

class CartPage extends StatefulWidget {
  const CartPage({super.key});

  @override
  State<CartPage> createState() => _CartPageState();
}

class _CartPageState extends State<CartPage> {
  @override
  Widget build(BuildContext context) {
    final cartBox = Hive.box("cart_box");

    deleteCart(String key) async {
      await cartBox.delete(key);
      setState(() {
        // Refresh the cart data
      });
    }

    var cartProvider = Provider.of<CartProvider>(context);
    cartProvider.getCart();

    return Scaffold(
      backgroundColor: const Color(0xFFE2E2E2),
      body: Stack(
        children: [
          Container(
            padding: const EdgeInsets.fromLTRB(16, 36, 0, 0),
            height: MediaQuery.of(context).size.height * 0.4,
            width: MediaQuery.of(context).size.width,
            decoration: const BoxDecoration(
              image: DecorationImage(
                  image: AssetImage("assets/images/top_image.png"),
                  fit: BoxFit.fill),
            ),
          ),
          Padding(
            padding: const EdgeInsets.only(left: 8, right: 8, top: 16),
            child: SingleChildScrollView(
              child: Column(

                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  const SizedBox(
                    height: 20,
                  ),
                  Padding(
                    padding: const EdgeInsets.only(left: 12),
                    child: Text(
                      "My Cart",
                      style: appStyle(36, Colors.white, FontWeight.bold),
                    ),
                  ),
```

```
const SizedBox(
 height: 20,
),
SizedBox(
 height: MediaQuery.of(context).size.height * 0.68,
 child: ListView.builder(
  itemCount: cartProvider.cart.length,
  padding: EdgeInsets.zero,
  itemBuilder: (context, index) {
   final data = cartProvider.cart[index];
   return Padding(
    padding: const EdgeInsets.only(left: 12, right: 12, bottom: 20),
    child: ClipRRect(
     borderRadius:
       const BorderRadius.all(Radius.circular(12)),
     child: Slidable(
      key: ValueKey(data['key']),
      endActionPane: ActionPane(
       motion: const ScrollMotion(),
       children: [
        SlidableAction(
         onPressed: (BuildContext context) {
          cartProvider.deleteItem(data[
            'key']); // Call the delete method from provider
         },
         backgroundColor: const Color(0xFF000000),
         foregroundColor: Colors.white,
         icon: Icons.delete,
         label: 'Delete',
        )
       ],
      ),
      child: Container(
       height:
         MediaQuery.of(context).size.height * 0.13,
       width: MediaQuery.of(context).size.width,
       decoration: BoxDecoration(
        color: Colors.grey.shade100,
        boxShadow: [
         BoxShadow(
           color: Colors.grey.shade500,
           spreadRadius: 5,
           blurRadius: 0.3,
           offset: const Offset(0, 1)),
        ],
       ),
       child: Row(
        mainAxisAlignment:
          MainAxisAlignment.spaceBetween,
        children: [
         Row(
          children: [
           Column(
            mainAxisAlignment:
              MainAxisAlignment.spaceBetween,
            crossAxisAlignment:
              CrossAxisAlignment.start,
            children: [
             Padding(
```

```
                padding:
                  const EdgeInsets.all(8),
                child: CachedNetworkImage(
                 imageUrl: data["imageUrl"],
                 width: 70,
                 height: 62,
                 fit: BoxFit.contain,
                ),
               ),
              ],
             ),
             Padding(
              padding: const EdgeInsets.only(
                top: 12, left: 20),
              child: Column(
               mainAxisAlignment:
                 MainAxisAlignment.start,
               crossAxisAlignment:
                 CrossAxisAlignment.start,
               children: [
                Text(
                 data['name'],
                 style: appStyle(
                   16,
                   Colors.black,
                   FontWeight.bold),
                ),
                const SizedBox(
                 height: 5,
                ),
                Text(
                 data['category'],
                 style: appStyle(
                   14,
                   Colors.grey,
                   FontWeight.w600),
                ),
                const SizedBox(
                 height: 5,
                ),
                Row(
                 children: [
                  Text(
                   data['price'],
                   style: appStyle(
                     18,
                     Colors.black,
                     FontWeight.w600),
                  ),
                  const SizedBox(
                   width: 40,
                  ),
                  Text(
                   "Size: 7",
                   style: appStyle(
                    18,
                    Colors.grey,
                    FontWeight.w600,
                   ),
```

```
                                ),
                              ],
                            ),
                          ],
                        ),
                      )
                    ],
                  ),
                  Row(
                    children: [
                      Padding(
                        padding: const EdgeInsets.all(8.0),
                        child: Container(
                          decoration: const BoxDecoration(
                              color: Colors.white,
                              borderRadius:
                                  BorderRadius.all(
                                      Radius.circular(16))),
                          child: Column(
                            mainAxisAlignment:
                                MainAxisAlignment
                                    .spaceBetween,
                            children: [
                              InkWell(
                                onTap: () {
                                  cartProvider
                                      .incrementQuantity(
                                          data['key']);
                                },
                                child: const Icon(
                                  AntDesign.plussquare,
                                  size: 20,
                                  color: Colors.black,
                                ),
                              ),
                              Text(
                                data['qty'].toString(),
                                style: appStyle(
                                  16,
                                  Colors.black,
                                  FontWeight.w600,
                                ),
                              ),
                              InkWell(
                                  onTap: () {
                                    cartProvider
                                        .decrementQuantity(
                                            data['key']);
                                  },
                                  child: const Icon(
                                    AntDesign.minussquare,
                                    size: 20,
                                    color: Colors.grey,
                                  )),
                            ],
                          ),
                        ),
                      ),
                    ),
```

```
                                // ),
                              ],
                            )
                          ],
                        ),
                      ),
                    ),
                  ),
                );
              },
            ),
          ),
          Align(
            alignment: Alignment.bottomCenter,
            child: CheckoutButton(
              label: "Proceed to Checkout",
              onTap: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) => const CheckoutPage()),
                );
              },
            ),
          ),
        ],
      ),
    ),
  ),
 );
 }

 void doNothing(BuildContext context) {}
}
```

Checkout_page.dart

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:sneaker_app/views/pages/proceed_page.dart';
import '../../controllers/cart_provider.dart';

class CheckoutPage extends StatefulWidget {
 const CheckoutPage({super.key});

 @override
 State<CheckoutPage> createState() => _CheckoutPageState();
}

class _CheckoutPageState extends State<CheckoutPage> {
 final _nameController = TextEditingController();
 final _phoneController = TextEditingController();
 final _addressController = TextEditingController();

 @override
 void dispose() {
  _nameController.dispose();
```

```
  _phoneController.dispose();
  _addressController.dispose();
  super.dispose();
 }

 @override
 Widget build(BuildContext context) {
  final cartProvider = Provider.of<CartProvider>(context);
  double subtotal = cartProvider.cart.fold(
   0,
   (sum, item) => sum + double.parse(item['price']) * item['qty'],
  );
  const double shippingPrice = 30.0;
  double totalPrice = subtotal + shippingPrice;

  return Scaffold(
   backgroundColor: const Color(0xFFE2E2E2),
   appBar: AppBar(
    backgroundColor: const Color(0xFFE2E2E2),
    title: const Text(
     "Checkout",
     style: TextStyle(fontSize: 26),
    ),
    centerTitle: true,
   ),
   body: SingleChildScrollView(
    padding: const EdgeInsets.all(16.0),
    child: Column(
     crossAxisAlignment: CrossAxisAlignment.start,
     children: [
      const Text(
       "Contact Information",
       style: TextStyle(
        fontSize: 22,
        fontWeight: FontWeight.bold,
       ),
      ),
      const SizedBox(height: 10),
      TextField(
       controller: _nameController,
       decoration: InputDecoration(
        labelText: "Enter Name", focusColor: Colors.black,
        labelStyle:
            const TextStyle(color: Colors.black54), // Label color
        border: OutlineInputBorder(
         borderRadius: BorderRadius.circular(12),
         borderSide: const BorderSide(color: Colors.black54),
        ),
        focusedBorder: OutlineInputBorder(
         borderRadius: BorderRadius.circular(12),
         borderSide: const BorderSide(color: Colors.black54),
        ),
        enabledBorder: OutlineInputBorder(
         borderRadius: BorderRadius.circular(12),
         borderSide: const BorderSide(color: Colors.black54),
        ),
       ),
       cursorColor: Colors.black54,
      ),
```

```
const SizedBox(height: 20),
TextField(
  controller: _phoneController,
  keyboardType: TextInputType.phone,
  decoration: InputDecoration(
    labelText: "Enter Phone",
    focusColor: Colors.black,
    labelStyle:
        const TextStyle(color: Colors.black54), // Label color
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: const BorderSide(color: Colors.black54),
    ),
    focusedBorder: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: const BorderSide(color: Colors.black54),
    ),
    enabledBorder: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: const BorderSide(color: Colors.black54),
    ),
    // cursorColor: Colors.black, // Cursor color
  ),
  cursorColor: Colors.black54,
),
const SizedBox(height: 20),
TextField(
  controller: _addressController,
  decoration: InputDecoration(
    labelText: "Enter Address",
    focusColor: Colors.black,
    labelStyle:
        const TextStyle(color: Colors.black54), // Label color
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: const BorderSide(color: Colors.black54),
    ),
    focusedBorder: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: const BorderSide(color: Colors.black54),
    ),
    enabledBorder: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: const BorderSide(color: Colors.black54),
    ),
  ),
  cursorColor: Colors.black54,
),
const SizedBox(height: 30),
const Text(
  "Order Summary",
  style: TextStyle(
    fontSize: 22,
    fontWeight: FontWeight.bold,
  ),
),
const SizedBox(height: 20),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
```

```
    children: [
     const Text(
      "Subtotal:",
      style: TextStyle(fontSize: 16),
     ),
     Text(
      "\₹${subtotal.toStringAsFixed(2)}",
      style: const TextStyle(fontSize: 16),
     ),
    ],
   ),
   const SizedBox(height: 20),
   Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
     const Text(
      "Shipping Price:",
      style: TextStyle(fontSize: 16),
     ),
     Text(
      "\₹${shippingPrice.toStringAsFixed(2)}",
      style: const TextStyle(fontSize: 16),
     ),
    ],
   ),
   const Divider(height: 30, thickness: 2),
   Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
     const Text(
      "Total:",
      style: TextStyle(
       fontSize: 20,
       fontWeight: FontWeight.bold,
      ),
     ),
     Text(
      "\₹${totalPrice.toStringAsFixed(2)}",
      style: const TextStyle(
       fontSize: 20,
       fontWeight: FontWeight.bold,
      ),
     ),
    ],
   ),
   const SizedBox(height: 40),
   SizedBox(
    width: double.infinity,
    child: ElevatedButton(
     onPressed: () {
      if (_nameController.text.isEmpty ||
        _phoneController.text.isEmpty ||
        _addressController.text.isEmpty) {
       // Show an error message if any field is empty
       ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
         content: Text("Please fill in all the fields."),
         backgroundColor: Colors.red,
         behavior: SnackBarBehavior.floating,
```

```
                    ),
                  );
                } else {
                  // Proceed to the next page if all fields are filled
                  Navigator.pushReplacement(
                    context,
                    MaterialPageRoute(
                      builder: (context) =>  ProceedPage(phone: _phoneController.text),
                    ),
                  );
                }
              },
              style: ElevatedButton.styleFrom(
                padding: const EdgeInsets.symmetric(vertical: 16),
                backgroundColor: Colors.black,
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(26),
                ),
              ),
              child: const Text(
                "Proceed",
                style: TextStyle(fontSize: 20, color: Colors.white),
              ),
            ),
          ),
        ),
      ],
    ),
  ),
 );
 }
}
```

favorites.dart

```
import 'package:cached_network_image/cached_network_image.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter_vector_icons/flutter_vector_icons.dart';
import 'package:provider/provider.dart';
import 'package:sneaker_app/views/shared/app_style.dart';
import '../../controllers/favorites_provider.dart';
import 'mainscreen.dart';

class Favourites extends StatefulWidget {
 const Favourites({super.key});

 @override
 State<Favourites> createState() => _FavouritesState();
}

class _FavouritesState extends State<Favourites> {
 final userEmail = FirebaseAuth.instance.currentUser?.email ?? '';

 @override
 Widget build(BuildContext context) {
  var favouritesNotifier =
      Provider.of<FavouritesNotifier>(context, listen: true);
  favouritesNotifier.getAllData();
```

```
return Scaffold(
  backgroundColor: const Color(0xFFE2E2E2),
  body: SizedBox(
    height: MediaQuery.of(context).size.height,
    width: MediaQuery.of(context).size.width,
    child: Stack(
      children: [
        Container(
          padding: const EdgeInsets.fromLTRB(16, 36, 0, 0),
          height: MediaQuery.of(context).size.height * 0.4,
          width: MediaQuery.of(context).size.width,
          decoration: const BoxDecoration(
            image: DecorationImage(
                image: AssetImage("assets/images/top_image.png"),
                fit: BoxFit.fill),
          ),
          child: Padding(
            padding: const EdgeInsets.all(8),
            child: Text(
              "My Favourites",
              style: appStyle(36, Colors.white, FontWeight.bold),
            ),
          ),
        ),
        Padding(
          padding: const EdgeInsets.all(8),
          child: ListView.builder(
            itemCount: favouritesNotifier.fav.length,
            padding: const EdgeInsets.only(top: 100),
            itemBuilder: (context, index) {
              final shoe = favouritesNotifier.fav[index];
              return Padding(
                padding: const EdgeInsets.all(8),
                child: ClipRRect(
                  borderRadius: const BorderRadius.all(
                    Radius.circular(12),
                  ),
                  child: Container(
                    height: MediaQuery.of(context).size.height * 0.136,
                    width: MediaQuery.of(context).size.width,
                    decoration: BoxDecoration(
                        color: Colors.grey.shade100,
                        boxShadow: [
                          BoxShadow(
                            color: Colors.grey.shade500,
                            spreadRadius: 5,
                            blurRadius: 0.3,
                            offset: const Offset(0, 1),
                          )
                        ]),
                    child: Row(
                      mainAxisAlignment: MainAxisAlignment.spaceBetween,
                      children: [
                        Row(
                          children: [
                            Padding(
                              padding: const EdgeInsets.all(10),
                              child: CachedNetworkImage(
                                imageUrl: shoe["imageUrl"],
```

```
                width: 70,
                height: 70,
              ),
            ),
            Padding(
              padding:
                  const EdgeInsets.only(top: 8, left: 8),
              child: Column(
                crossAxisAlignment:
                    CrossAxisAlignment.start,
                children: [
                  Text(
                    shoe["name"],
                    style: appStyle(
                        16, Colors.black, FontWeight.bold),
                  ),
                  const SizedBox(
                    height: 3,
                  ),
                  Text(
                    shoe["category"],
                    style: appStyle(
                        14, Colors.grey, FontWeight.w600),
                  ),
                  const SizedBox(
                    height: 3,
                  ),
                  Row(
                    mainAxisAlignment:
                        MainAxisAlignment.spaceBetween,
                    children: [
                      Text(
                        "${shoe["price"]}",
                        style: appStyle(18, Colors.black,
                            FontWeight.w600),
                      ),
                    ],
                  )
                ],
              ),
            )
          ],
        ),
        Padding(
          padding: const EdgeInsets.all(8),
          child: GestureDetector(
            onTap: () {
              favouritesNotifier.deleteFav(shoe["key"]);
              favouritesNotifier.ids.removeWhere(
                  (element) => element == shoe['id']);
              Navigator.push(
                  context,
                  MaterialPageRoute(
                      builder: (context) => MainScreen(userEmail: userEmail)));
            },
            child: const Icon(Ionicons.md_heart_dislike),
          ),
        )
      ],
```

```
                    ),
                  ),
                ),
              );
            },
          ),
        )
      ],
    ),
  ),
 );
 }
}
```

home_page.dart

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:sneaker_app/views/shared/app_style.dart';

import '../../controllers/product_provider.dart';
import '../shared/home_widget.dart';

class HomePage extends StatefulWidget {
 const HomePage({super.key});

 @override
 State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> with TickerProviderStateMixin {
 late final TabController _tabController =
    TabController(length: 3, vsync: this);

 @override
 Widget build(BuildContext context) {
  var productNotifier = Provider.of<ProductNotifier>(context);
  productNotifier.getMale();
  productNotifier.getFemale();
  productNotifier.getkids();

  return Scaffold(
   backgroundColor: const Color(0xFFE2E2E2),
   body: SizedBox(
    height: MediaQuery.of(context).size.height,
    child: Stack(
     children: [
      Container(
       padding: const EdgeInsets.fromLTRB(16, 36, 0, 0),
       height: MediaQuery.of(context).size.height * 0.4,
       decoration: const BoxDecoration(
        image: DecorationImage(
          image: AssetImage("assets/images/top_image.png"),
          fit: BoxFit.fill),
       ),
       child: Container(
        padding: const EdgeInsets.only(left: 8, bottom: 15),
        width: MediaQuery.of(context).size.width,
        child: Column(
```

```
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text(
                    "SneakRush",
                    style: appstyleWithHt(
                        38, Colors.white, FontWeight.bold, 1.5),
                  ),
                  Text(
                    "Collection",
                    style: appstyleWithHt(
                        38, Colors.white, FontWeight.bold, 1.2),
                  ),
                  TabBar(
                    dividerColor: Colors.transparent,
                    padding: EdgeInsets.zero,
                    indicatorSize: TabBarIndicatorSize.label,
                    indicatorColor: Colors.transparent,
                    controller: _tabController,
                    isScrollable: true,
                    labelColor: Colors.white,
                    labelStyle: appStyle(20, Colors.white, FontWeight.bold),
                    unselectedLabelColor: Colors.grey.withOpacity(0.6),
                    tabs: const [
                      Tab(
                        text: "Men Shoes",
                      ),
                      Tab(
                        text: "Women Shoes",
                      ),
                      Tab(
                        text: "Kids Shoes",
                      )
                    ],
                  ),
                ],
              ),
            ),
          ),
          Padding(
            padding: EdgeInsets.only(
                top: MediaQuery.of(context).size.height * 0.275),
            child: Padding(
              padding: const EdgeInsets.only(left: 12),
              child: TabBarView(
                controller: _tabController,
                children: [
                  HomeWidget(male: productNotifier.male, tabIndexNo: 0),
                  HomeWidget(male: productNotifier.female, tabIndexNo: 1),
                  HomeWidget(male: productNotifier.kids, tabIndexNo: 2),
                ],
              ),
            ),
          )
        ],
      ),
    ),
  );
 }
}
```

login_signup_page.dart

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:sneaker_app/views/pages/mainscreen.dart';

class LoginSignupScreen extends StatefulWidget {
  const LoginSignupScreen({super.key});

  @override
  _LoginSignupScreenState createState() => _LoginSignupScreenState();
}

class _LoginSignupScreenState extends State<LoginSignupScreen> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  final TextEditingController confirmPasswordController = TextEditingController();

  bool isSignup = false;
  bool isLoading = false;

  @override
  void dispose() {
    emailController.dispose();
    passwordController.dispose();
    confirmPasswordController.dispose();
    super.dispose();
  }

  void submit() async {
    if (_formKey.currentState!.validate()) {
      _formKey.currentState!.save();
      setState(() => isLoading = true);

      try {
        UserCredential userCredential;
        if (isSignup) {
          if (passwordController.text.trim() != confirmPasswordController.text.trim()) {
            showErrorMessage('Passwords do not match');
            return;
          }
          userCredential = await _auth.createUserWithEmailAndPassword(
            email: emailController.text.trim(),
            password: passwordController.text.trim(),
          );
        } else {
          userCredential = await _auth.signInWithEmailAndPassword(
            email: emailController.text.trim(),
            password: passwordController.text.trim(),
          );
        }
        // Navigate to MainScreen and pass email
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(
            builder: (context) => MainScreen(userEmail: userCredential.user?.email ?? ''),
```

```
        ),
       );
     } on FirebaseAuthException catch (e) {
       // Handle errors
       showErrorMessage(e.message ?? 'An error occurred. Please try again later.');
     } finally {
       setState(() => isLoading = false);
     }
   }
 }

 void showErrorMessage(String message) {
   ScaffoldMessenger.of(context).showSnackBar(
     SnackBar(
       content: Text(message),
       backgroundColor: Colors.red,
     ),
   );
 }

 @override
 Widget build(BuildContext context) {
   return Scaffold(
     resizeToAvoidBottomInset: false,
     body: Stack(
       children: [
         // Background Image
         Positioned.fill(
           child: Image.asset(
             'assets/images/background.png',
             fit: BoxFit.cover,
           ),
         ),
         // Login/Signup Form
         Padding(
           padding: const EdgeInsets.all(24),
           child: Column(
             mainAxisAlignment: MainAxisAlignment.center,
             crossAxisAlignment: CrossAxisAlignment.stretch,
             children: [
               Text(
                 isSignup ? 'Signup' : 'Login',
                 textAlign: TextAlign.center,
                 style: const TextStyle(
                   color: Colors.white,
                   fontSize: 28,
                   fontWeight: FontWeight.bold,
                 ),
               ),
               const SizedBox(height: 20),
               Form(
                 key: _formKey,
                 child: Column(
                   children: [
                     buildTextField(
                       'Email',
                       controller: emailController,
                     ),
                     const SizedBox(height: 12),
```

```
            buildTextField(
              'Password',
              controller: passwordController,
              obscureText: true,
            ),
            if (isSignup)
              const SizedBox(height: 12),
            if (isSignup)
              buildTextField(
                'Confirm Password',
                controller: confirmPasswordController,
                obscureText: true,
              ),
          ],
        ),
      ),
      const SizedBox(height: 20),
      if (isLoading)
        const Center(
          child: CircularProgressIndicator(color: Colors.white),
        ),
      if (!isLoading)
        GestureDetector(
          onTap: submit,
          child: Container(
            decoration: BoxDecoration(
              color: const Color.fromARGB(255, 137, 137, 137),
              // gradient: LinearGradient(
              //   colors: [Colors.grey[800]!, const Color.fromARGB(255, 99, 99, 99)!],
              //   begin: Alignment.topLeft,
              //   end: Alignment.bottomRight,
              // ),
              borderRadius: BorderRadius.circular(8),
            ),
            padding: const EdgeInsets.symmetric(
                horizontal: 30, vertical: 15),
            child: Text(
              isSignup ? 'Signup' : 'Login',
              textAlign: TextAlign.center,
              style: const TextStyle(
                color: Colors.white,
                fontSize: 16,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
        ),
      if (!isSignup)
        Align(
          alignment: Alignment.centerRight,
          child: TextButton(
            onPressed: () {},
            child: const Text(
              'Forgot Password?',
              style: TextStyle(color: Colors.white),
            ),
          ),
        ),
      Row(
```

```
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  Text(
                    isSignup
                        ? 'Already have an account? '
                        : 'Don't have an account? ',
                    style: const TextStyle(
                      color: Colors.grey,
                      fontSize: 16,
                    ),
                  ),
                  TextButton(
                    onPressed: () {
                      setState(() => isSignup = !isSignup);
                    },
                    child: Text(
                      isSignup ? 'Login' : 'Signup',
                      style: const TextStyle(color: Colors.white),
                    ),
                  ),
                ],
              ),
            ],
          ),
        ),
      ],
    ),
  );
}

Widget buildTextField(
  String labelText, {
  required TextEditingController controller,
  bool obscureText = false,
}) {
  return TextFormField(
    controller: controller,
    decoration: InputDecoration(
      focusedBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
        borderSide:
            BorderSide(color: Colors.grey[400]!), // Light grey when focused
      ),
      enabledBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
        borderSide: BorderSide(
            color: Colors.grey[500]!), // Darker grey when not focused
      ),
      labelText: labelText,
      labelStyle: const TextStyle(color: Colors.white),
      filled: true,
      fillColor: Colors.grey[700], // Dark gray for background
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
        borderSide: const BorderSide(color: Colors.white),
      ),
    ),
    style: const TextStyle(color: Colors.white),
    obscureText: obscureText,
```

```
      cursorColor: const Color.fromARGB(255, 202, 202, 202),
      validator: (value) {
        if (value!.isEmpty) return 'Please enter $labelText';
        if (labelText == 'Email' &&
            !RegExp(r'^[^@]+@[^@]+\.[^@]+').hasMatch(value)) {
          return 'Enter a valid email';
        }
        return null;
      },
    );
  }
}
```

Mainscreen.dart

```
import 'package:flutter/material.dart';
import 'package:sneaker_app/views/pages/cart_page.dart';
import 'package:sneaker_app/views/pages/home_page.dart';
import 'package:sneaker_app/views/pages/profile_page.dart';

// SERVICES
import '../../controllers/mainscreen_provider.dart';
import 'package:provider/provider.dart';

// PAGES
import '../shared/bottom_nav.dart';

import 'favorites.dart';

class MainScreen extends StatelessWidget {
  final String userEmail;

  const MainScreen({super.key, required this.userEmail});

  List<Widget> pageList(String userEmail) {
    return [
      const HomePage(),
      const Favourites(),
      const CartPage(),
      ProfilePage(userEmail: userEmail),
    ];
  }

  @override
  Widget build(BuildContext context) {
    return Consumer<MainScreenNotifier>(
      builder: (context, mainScreenNotifier, child) {
        return Scaffold(
          backgroundColor: const Color(0xFFE2E2E2),
          body: pageList(userEmail)[mainScreenNotifier.pageIndex],
          bottomNavigationBar: const BottomNav(),
        );
      },
    );
  }
}
```

Proceed_page.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:lottie/lottie.dart';
import 'package:provider/provider.dart';
import 'package:sneaker_app/controllers/mainscreen_provider.dart';
import 'package:sneaker_app/views/pages/mainscreen.dart';

class ProceedPage extends StatelessWidget {
   final userEmail = FirebaseAuth.instance.currentUser?.email ?? '';

  final String phone;

   ProceedPage({super.key, required this.phone});
  @override
  Widget build(BuildContext context) {
   return Scaffold(
     backgroundColor: const Color(0xFFE2E2E2),
     appBar: AppBar(
       backgroundColor: const Color(0xFFE2E2E2),
       title: const Text("Order Confirmation"),
       centerTitle: true,
     ),
     body: Padding(
       padding: const EdgeInsets.all(16.0),
       child: Column(
         mainAxisAlignment: MainAxisAlignment.start,
         crossAxisAlignment: CrossAxisAlignment.center,
         children: [
           // Lottie animation
           Center(
             child: Lottie.asset(
               'assets/animation/success.json',
               width: 400,
               height: 300,
               repeat: false,
             ),
           ),
           const SizedBox(height: 20),
           // Order confirmation text
           const Text(
             "Your order is confirmed",
             style: TextStyle(
               fontSize: 28,
               fontWeight: FontWeight.bold,
             ),
             textAlign: TextAlign.center,
           ),
           const SizedBox(height: 20),
           // Information text
           Text(
             "We have sent order confirmation and other details to $phone.",
             style: const TextStyle(fontSize: 16),
             textAlign: TextAlign.center,
           ),
           const SizedBox(height: 40),
           // Continue Shopping Button
           SizedBox(
             width: double.infinity,
             child: ElevatedButton(
```

```
            onPressed: () {
              // Set the page index to Home (0)
              context.read<MainScreenNotifier>().pageIndex = 0;

              // Navigate back to MainScreen and ensure it doesn't add to the stack
              Navigator.pushAndRemoveUntil(
                context,
                MaterialPageRoute(builder: (context) => MainScreen(userEmail:
userEmail)),
                (route) => false, // Clear the navigation stack
              );
            },
            style: ElevatedButton.styleFrom(
              padding: const EdgeInsets.symmetric(vertical: 16),
              backgroundColor: Colors.black,
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(26),
              ),
            ),
            child: const Text(
              "Continue Shopping",
              style: TextStyle(fontSize: 20, color: Colors.white),
            ),
          ),
        ),
      ],
    ),
  ),
 );
 }
}
```

Product_by_cat.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_vector_icons/flutter_vector_icons.dart';
import 'package:provider/provider.dart';
import 'package:sneaker_app/views/shared/app_style.dart';
import 'package:sneaker_app/views/shared/category_button.dart';
import '../../controllers/product_provider.dart';
import '../shared/custom_spacer.dart';
import '../shared/latest_shoes.dart';

class ProductByCategory extends StatefulWidget {
 const ProductByCategory({super.key, required this.tabIndex});

 final int tabIndex;

 @override
 State<ProductByCategory> createState() => _ProductByCategoryState();
}

class _ProductByCategoryState extends State<ProductByCategory>
   with TickerProviderStateMixin {
 late TabController _tabController;
 double value = 100; // Move value here for persistent state

 @override
 void initState() {
```

```
  super.initState();
  _tabController = TabController(length: 3, vsync: this);
  _tabController.animateTo(widget.tabIndex, curve: Curves.easeIn);
 }

 @override
 void dispose() {
  _tabController.dispose();
  super.dispose();
 }

 List<String> brand = [
  "assets/images/adidas.png",
  "assets/images/nike.png",
  "assets/images/gucci.png",
  "assets/images/jordan.png",
 ];

 @override
 Widget build(BuildContext context) {
  var productNotifier = Provider.of<ProductNotifier>(context);
  productNotifier.getMale();
  productNotifier.getFemale();
  productNotifier.getkids();
  return Scaffold(
   backgroundColor: const Color(0xFFE2E2E2),
   body: SizedBox(
    height: MediaQuery.of(context).size.height,
    child: Stack(
     children: [
      Container(
       padding: const EdgeInsets.fromLTRB(16, 45, 0, 0),
       height: MediaQuery.of(context).size.height * 0.4,
       decoration: const BoxDecoration(
        image: DecorationImage(
         image: AssetImage("assets/images/top_image.png"),
         fit: BoxFit.fill,
        ),
       ),
       child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
         Padding(
          padding: const EdgeInsets.fromLTRB(6, 12, 16, 18),
          child: Row(
           mainAxisAlignment: MainAxisAlignment.spaceBetween,
           children: [
            GestureDetector(
             onTap: () {
              Navigator.pop(context);
             },
             child: const Icon(
              AntDesign.close,
              color: Colors.white,
             ),
            ),
            GestureDetector(
             onTap: () {
              filter();
```

```
                    },
                    child: const Icon(
                      FontAwesome.sliders,
                      color: Colors.white,
                    ),
                  ),
                ],
              ),
            ),
            TabBar(
              dividerColor: Colors.transparent,
              padding: EdgeInsets.zero,
              indicatorSize: TabBarIndicatorSize.label,
              indicatorColor: Colors.transparent,
              controller: _tabController,
              isScrollable: true,
              labelColor: Colors.white,
              labelStyle: appStyle(24, Colors.white, FontWeight.bold),
              unselectedLabelColor: Colors.grey.withOpacity(0.3),
              tabs: const [
                Tab(
                  text: "Men Shoes",
                ),
                Tab(
                  text: "Women Shoes",
                ),
                Tab(
                  text: "Kids Shoes",
                ),
              ],
            ),
          ],
        ),
      ),
      Padding(
        padding: EdgeInsets.only(
          top: MediaQuery.of(context).size.height * 0.235,
          left: 16,
          right: 12,
        ),
        child: ClipRRect(
          borderRadius: const BorderRadius.all(Radius.circular(16)),
          child: TabBarView(
            controller: _tabController,
            children: [
              LatestShoes(male: productNotifier.male),
              LatestShoes(male: productNotifier.female),
              LatestShoes(male: productNotifier.kids),
            ],
          ),
        ),
      ),
    ],
  ),
 ),
);
}

Future<dynamic> filter() {
```

```
return showModalBottomSheet(
  context: context,
  isScrollControlled: true,
  backgroundColor: Colors.transparent,
  barrierColor: Colors.white54,
  builder: (context) {
   double modalValue = value; // Local value scoped to modal

   return StatefulBuilder(
    builder: (BuildContext context, StateSetter modalSetState) {
     return Container(
      height: MediaQuery.of(context).size.height * 0.84,
      decoration: const BoxDecoration(
       color: Colors.white,
       borderRadius: BorderRadius.only(
        topLeft: Radius.circular(25),
        topRight: Radius.circular(25),
       ),
      ),
      child: SingleChildScrollView(
       child: Column(
        children: [
         const SizedBox(height: 10),
         Container(
          height: 5,
          width: 40,
          decoration: const BoxDecoration(
           borderRadius: BorderRadius.all(Radius.circular(10)),
           color: Colors.black38,
          ),
         ),
         const SizedBox(height: 20),
         Text(
          "Filter",
          style: appStyle(40, Colors.black, FontWeight.bold),
         ),
         const CustomSpacer(),
         Text(
          "Gender",
          style: appStyle(20, Colors.black, FontWeight.bold),
         ),
         const SizedBox(height: 20),
         const Row(
          mainAxisAlignment: MainAxisAlignment.spaceAround,
          children: [
           CategoryBtn(buttonClr: Colors.black, label: "Men"),
           CategoryBtn(buttonClr: Colors.grey, label: "Women"),
           CategoryBtn(buttonClr: Colors.grey, label: "Kids"),
          ],
         ),
         const CustomSpacer(),
         Text(
          "Category",
          style: appStyle(20, Colors.black, FontWeight.w600),
         ),
         const SizedBox(height: 20),
         const Row(
          mainAxisAlignment: MainAxisAlignment.spaceAround,
          children: [
```

```
            CategoryBtn(label: "Shoes", buttonClr: Colors.black),
            CategoryBtn(label: "Apparel", buttonClr: Colors.grey),
            CategoryBtn(label: "Sports", buttonClr: Colors.grey),
          ],
        ),
        const CustomSpacer(),
        Text(
          "Price",
          style: appStyle(20, Colors.black, FontWeight.bold),
        ),
        const CustomSpacer(),
        Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              Text(
                '\₹${modalValue.toStringAsFixed(2)}',
                style: const TextStyle(fontSize: 16),
              ),
              Slider(
                value: modalValue,
                activeColor: Colors.black,
                inactiveColor: Colors.grey,
                thumbColor: Colors.black,
                max: 500,
                label: modalValue.round().toString(),
                onChanged: (double newValue) {
                  modalSetState(() {
                    modalValue = newValue; // Update locally in modal
                  });

                  // Update global value as well
                  setState(() {
                    value = newValue;
                  });
                },
              ),
            ],
          ),
        ),
        const CustomSpacer(),
        Text(
          "Brand",
          style: appStyle(20, Colors.black, FontWeight.bold),
        ),
        const SizedBox(height: 20),
        Container(
          padding: const EdgeInsets.all(8),
          height: 80,
          child: ListView.builder(
            itemCount: brand.length,
            scrollDirection: Axis.horizontal,
            itemBuilder: (context, index) {
              return Padding(
                padding: const EdgeInsets.all(8),
                child: Container(
                  height: 60,
                  padding: const EdgeInsets.all(8),
                  decoration: BoxDecoration(
```

```
                        color: Colors.grey.shade200,
                        borderRadius: const BorderRadius.all(
                          Radius.circular(12),
                        ),
                      ),
                      child: Image.asset(
                        brand[index],
                        height: 60,
                        width: 80,
                        color: Colors.black,
                      ),
                    ),
                  );
                },
              ),
            ),
          ],
        ),
      ),
    );
  },
 );
}

}
```

Product_page.dart

```
import 'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:flutter_rating_bar/flutter_rating_bar.dart';
import 'package:flutter_vector_icons/flutter_vector_icons.dart';
import 'package:hive/hive.dart';
import 'package:provider/provider.dart';
import 'package:sneaker_app/controllers/product_provider.dart';
import 'package:sneaker_app/views/shared/app_style.dart';
import '../../controllers/favorites_provider.dart';
import '../../models/constants.dart';
import '../../models/sneakers_model.dart';
import '../shared/checkout_btn.dart';
import 'favorites.dart';

class ProductPage extends StatefulWidget {
  const ProductPage({super.key, required this.id, required this.category});

  final String id;
  final String category;

  @override
  State<ProductPage> createState() => _ProductPageState();
}

class _ProductPageState extends State<ProductPage> {
  final PageController pageController = PageController();
  final _cartBox = Hive.box("cart_box");
  // final _favBox = Hive.box("fav_box");
```

```
Future<void> _createCart(Map<String, dynamic> newCart) async {
  await _cartBox.add(newCart);
}

@override
Widget build(BuildContext context) {
  var productNotifier = Provider.of<ProductNotifier>(context);
  productNotifier.getShoes(widget.category, widget.id);

  var favouritesNotifier =
      Provider.of<FavouritesNotifier>(context, listen: true);
  favouritesNotifier.getFavourites();

  return Scaffold(
    body: FutureBuilder<Sneakers>(
      future: productNotifier.sneaker,
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.waiting) {
          return const CircularProgressIndicator();
        } else if (snapshot.hasError) {
          return Text("ERROR: $snapshot.error");
        } else {
          final sneaker = snapshot.data;
          return Consumer<ProductNotifier>(
            builder: (context, productNotifier, child) {
              return CustomScrollView(
                slivers: [
                  SliverAppBar(
                    automaticallyImplyLeading: false,
                    leadingWidth: 0,
                    title: Padding(
                      padding: const EdgeInsets.only(bottom: 10),
                      child: Row(
                        mainAxisAlignment: MainAxisAlignment.spaceBetween,
                        children: [
                          GestureDetector(
                            onTap: () {
                              Navigator.pop(context);
                              productNotifier.shoeSizes.clear();
                            },
                            child: const Icon(
                              AntDesign.close,
                              color: Colors.black,
                            ),
                          ),
                        ],
                      ),
                    ),
                    pinned: true,
                    snap: false,
                    floating: true,
                    backgroundColor: Colors.transparent,
                    expandedHeight: MediaQuery.of(context).size.height,
                    flexibleSpace: FlexibleSpaceBar(
                      background: Stack(
                        children: [
                          Container(
                            height: MediaQuery.of(context).size.height * 0.39,
                            width: MediaQuery.of(context).size.width,
```

```
color: Colors.grey.shade300,
child: Padding(
 padding: const EdgeInsets.all(16),
 child: CachedNetworkImage(
  imageUrl: sneaker!
     .imageUrl[0], // Show the first image only
  fit: BoxFit.contain,
 ),
),
),
Positioned(
 top: MediaQuery.of(context).size.height * 0.06,
 right: 16,
 child: Consumer<FavouritesNotifier>(
  builder: (context, favouritesNotifier, child) {
   return GestureDetector(
    onTap: () async {
     if (favouritesNotifier.ids
        .contains(sneaker.id)) {
      Navigator.push(
       context,
       MaterialPageRoute(
          builder: (context) =>
             const Favourites()),
      );
     } else {
      await favouritesNotifier.createFav({
       "id": sneaker.id,
       "name": sneaker.name,
       "category": sneaker.category,
       "price": sneaker.price,
       "imageUrl": sneaker.imageUrl[0],
      });
      setState(() {
       ids.add(sneaker
          .id); // Update the state to include the new favorite
      });
     }
    },
    child: favouritesNotifier.ids
         .contains(sneaker.id)
      ? const Icon(AntDesign.heart)
      : const Icon(AntDesign.hearto),
   );
  },
 ),
),
Positioned(
 bottom: 0,
 right: 0,
 left: 0,
 height: MediaQuery.of(context).size.height * 0.3,
 child: const Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
   Padding(
    padding:
       EdgeInsets.symmetric(horizontal: 4),
    child: CircleAvatar(
```

```
                radius: 5,
                backgroundColor: Colors
                    .black, // Highlight the current indicator
              ),
            ),
          ],
        ),
      ),
      Positioned(
        bottom: 30,
        child: ClipRRect(
          borderRadius: const BorderRadius.only(
            topLeft: Radius.circular(30),
            topRight: Radius.circular(30),
          ),
          child: Container(
            height:
                MediaQuery.of(context).size.height * 0.62,
            width: MediaQuery.of(context).size.width,
            color: Colors.white,
            child: Padding(
              padding: const EdgeInsets.all(12),
              child: SingleChildScrollView(
                child: Column(
                  crossAxisAlignment:
                      CrossAxisAlignment.start,
                  children: [
                    Text(
                      sneaker.name,
                      style: appStyle(34, Colors.black,
                          FontWeight.bold),
                    ),
                    Row(
                      children: [
                        Text(
                          sneaker.category,
                          style: appStyle(20, Colors.grey,
                              FontWeight.w500),
                        ),
                        const SizedBox(
                          width: 20,
                        ),
                        RatingBar.builder(
                          initialRating: 4,
                          minRating: 1,
                          direction: Axis.horizontal,
                          allowHalfRating: true,
                          itemCount: 5,
                          itemSize: 22,
                          itemPadding:
                              const EdgeInsets.symmetric(
                                  horizontal: 1),
                          itemBuilder: (context, _) =>
                              const Icon(
                            Icons.star,
                            size: 18,
                            color: Colors.black,
                          ),
                          onRatingUpdate: (rating) {},
```

```
                    )
                ],
              ),
              const SizedBox(
                height: 18,
              ),
              Row(
                mainAxisAlignment:
                    MainAxisAlignment.spaceBetween,
                children: [
                  Text(
                    "\₹${sneaker.price}",
                    style: appStyle(
                        26,
                        Colors.black,
                        FontWeight.w600),
                  ),
                  Row(
                    children: [
                      Text(
                        "Colors",
                        style: appStyle(
                            18,
                            Colors.black,
                            FontWeight.w500),
                      ),
                      const SizedBox(
                        width: 5,
                      ),
                      const CircleAvatar(
                        radius: 7,
                        backgroundColor:
                            Colors.black,
                      ),
                      const SizedBox(
                        width: 5,
                      ),
                      const CircleAvatar(
                        radius: 7,
                        backgroundColor:
                            Colors.grey,
                      ),
                      const SizedBox(
                        width: 8,
                      ),
                    ],
                  ),
                ],
              ),
              const SizedBox(
                height: 20,
              ),
              Column(
                children: [
                  Row(
                    crossAxisAlignment:
                        CrossAxisAlignment.end,
                    children: [
                      Text(
```

```
                "Select sizes",
                style: appStyle(
                    20,
                    Colors.black,
                    FontWeight.w700),
                ),
                const SizedBox(
                  width: 20,
                ),
                Text(
                  "View size guide",
                  style: appStyle(
                    16,
                    Colors.grey,
                    FontWeight.w600),
                ),
              ],
            ),
            const SizedBox(
              height: 10,
            ),
            SizedBox(
              height: 50,
              child: ListView.builder(
                itemCount: productNotifier
                    .shoeSizes.length,
                scrollDirection:
                    Axis.horizontal,
                padding: EdgeInsets.zero,
                itemBuilder:
                    (context, index) {
                  final sizes =
                      productNotifier
                          .shoeSizes[index];
                  return Padding(
                    padding: const EdgeInsets
                        .symmetric(
                        horizontal: 8),
                    child: ChoiceChip(
                      shape:
                          RoundedRectangleBorder(
                        borderRadius:
                            BorderRadius
                                .circular(60),
                        side:
                            const BorderSide(
                          color: Colors.black,
                          width: 1,
                          style: BorderStyle
                              .solid,
                        ),
                      ),
                      disabledColor:
                          Colors.white,
                      label: Text(
                        sizes["size"],
                        style: appStyle(
                            18,
                            sizes['isSelected']
```

```
                                ? Colors.white
                                : Colors
                                    .black,
                            FontWeight.w600),
                      ),
                      selectedColor:
                          Colors.black,
                      padding:
                          const EdgeInsets
                              .symmetric(
                              vertical: 8),
                      selected:
                          sizes["isSelected"],
                      onSelected: (newState) {
                        if (productNotifier
                            .sizes
                            .contains(sizes[
                                "size"])) {
                          productNotifier
                              .sizes
                              .remove(sizes[
                                  "size"]);
                        } else {
                          productNotifier
                              .sizes
                              .add(sizes[
                                  "size"]);
                        }
                        productNotifier
                            .toggleCheck(
                                index);
                        if (newState) {
                          // When size is selected
                          ScaffoldMessenger
                                  .of(context)
                              .showSnackBar(
                            SnackBar(
                              content: Text(
                                  "${sizes['size']} selected successfully!"),
                            ),
                          );
                        } else {
                          // When size is unselected
                          ScaffoldMessenger
                                  .of(context)
                              .showSnackBar(
                            SnackBar(
                              content: Text(
                                  "${sizes['size']} unselected successfully"),
                            ),
                          );
                        }
                      },
                      showCheckmark: false,
                    ),
                  );
                },
              ),
            ),
```

```
                ],
              ),
              const SizedBox(
                height: 10,
              ),
              const Divider(
                indent: 10,
                endIndent: 10,
                color: Colors.black,
              ),
              const SizedBox(
                height: 10,
              ),
              SizedBox(
                width: MediaQuery.of(context)
                    .size
                    .width *
                  0.8,
                child: Text(
                  sneaker.title,
                  maxLines: 2,
                  style: appStyle(24, Colors.black,
                    FontWeight.w700),
                ),
              ),
              const SizedBox(
                height: 10,
              ),
              Text(
                sneaker.description,
                textAlign: TextAlign.justify,
                maxLines: 6,
                style: appStyle(16, Colors.black,
                  FontWeight.normal),
              ),
              const SizedBox(
                height: 10,
              ),
              Align(
                alignment: Alignment.bottomCenter,
                child: Padding(
                  padding: const EdgeInsets.only(
                    top: 12),
                  child: CheckoutButton(
                    onTap: () async {
                      if (productNotifier
                          .sizes.isEmpty) {
                        // Show SnackBar if no size is selected
                        ScaffoldMessenger.of(
                            context)
                          .showSnackBar(
                          const SnackBar(
                            content: Text(
                              "Please select a size before adding to cart!"),
                          ),
                        );
                        return; // Stop further execution
                      }
                      ScaffoldMessenger.of(context)
```

```
                                        .showSnackBar(
                                      const SnackBar(
                                        content: Text(
                                            "Added to cart successfully!"),
                                      ),
                                    );
                                    _createCart(
                                      {
                                        "id": sneaker.id,
                                        "name": sneaker.name,
                                        "category":
                                            sneaker.category,
                                        "sizes":
                                            productNotifier.sizes,
                                        "imageUrl":
                                            sneaker.imageUrl[0],
                                        "price": sneaker.price,
                                        "qty": 1
                                      },
                                    );
                                    print(productNotifier.sizes);
                                    productNotifier.sizes.clear();
                                    // Navigator.pop(context);
                                  },
                                  label: "Add to Cart",
                                ),
                              ),
                            ),
                            const SizedBox(
                              height: 40,
                            ),
                          ],
                        ),
                      ),
                    ),
                  ),
                ),
              ),
            ],
          ),
        ),
      ],
    ),
  );
 }
}
```

Profile_page.dart

```
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:image_picker/image_picker.dart';
```

```
class ProfilePage extends StatefulWidget {
  final String userEmail;

  const ProfilePage({super.key, required this.userEmail});

  @override
  State<ProfilePage> createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
  String? name;
  String? address;
  String? phone;
  String? birthday;
  File? profileImage;

  final ImagePicker _picker = ImagePicker();

  void signOut(BuildContext context) async {
    await FirebaseAuth.instance.signOut();
    Navigator.pushReplacementNamed(context, '/login');
  }

  void navigateToFavorites() {
    Navigator.pushNamed(context, '/favorites');
  }

  void navigateToCart() {
    Navigator.pushNamed(context, '/cart');
  }

  void editField(String field, String title, void Function(String) onSave) {
    TextEditingController controller = TextEditingController();
    showDialog(
      context: context,
      builder: (context) => AlertDialog(
        backgroundColor: const Color(0xFFE2E2E2),
        title: Text("Edit $title"),
        content: TextField(
          controller: controller,
          decoration: InputDecoration(hintText: "Enter your $title"),
        ),
        actions: [
          TextButton(
            onPressed: () => Navigator.pop(context),
            child: const Text("Cancel", style: TextStyle(color: Colors.black)),
          ),
          ElevatedButton(
            style: ElevatedButton.styleFrom(backgroundColor: Colors.black,
foregroundColor: Colors.white),
            onPressed: () {
              onSave(controller.text);
              Navigator.pop(context);
            },
            child: const Text("Save"),
          ),
        ],
      ),
```

```
  );
 }

 void showImageOptions() {
  showDialog(
   context: context,
   builder: (context) => AlertDialog(
    backgroundColor: const Color(0xFFE2E2E2),
    title: const Text("Profile Picture"),
    content: Column(
     mainAxisSize: MainAxisSize.min,
     children: [
      ListTile(
       leading: const Icon(Icons.photo),
       title: const Text("Pick from Gallery"),
       onTap: pickImage,
      ),
      ListTile(
       leading: const Icon(Icons.delete),
       title: const Text("Remove Image"),
       onTap: profileImage != null ? removeImage : null,
       enabled: profileImage != null,
      ),
     ],
    ),
    actions: [
     TextButton(
      onPressed: () => Navigator.pop(context),
      child: const Text("Cancel",style: TextStyle(color: Colors.black),),
     ),
    ],
   ),
  );
 }

 Future<void> pickImage() async {
  Navigator.pop(context); // Close the dialog
  final XFile? pickedFile = await _picker.pickImage(source: ImageSource.gallery);
  if (pickedFile != null) {
   setState(() {
    profileImage = File(pickedFile.path);
   });
  }
 }

 void removeImage() {
  Navigator.pop(context); // Close the dialog
  setState(() {
   profileImage = null;
  });
 }

 @override
 Widget build(BuildContext context) {
  return SafeArea(
   child: Scaffold(
    backgroundColor: const Color(0xFFE2E2E2),
    body: Padding(
     padding: const EdgeInsets.all(16.0),
```

```
child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Center(
      child: Text(
        widget.userEmail,
        style: const TextStyle(fontSize: 26, fontWeight: FontWeight.bold),
      ),
    ),
    const SizedBox(height: 16),
    Center(
      child: GestureDetector(
        onTap: showImageOptions,
        child: CircleAvatar(
          radius: 50,
          backgroundColor: Colors.grey,
          backgroundImage: profileImage != null ? FileImage(profileImage!) : null,
          child: profileImage == null
              ? const Icon(Icons.person, size: 50, color: Colors.white)
              : null,
        ),
      ),
    ),
    const SizedBox(height: 24),
    buildEditableField(
      icon: Icons.person,
      label: "Name",
      value: name,
      onTap: () => editField(
          "Name", "name", (value) => setState(() => name = value)),
    ),
    buildEditableField(
      icon: Icons.location_on,
      label: "Address",
      value: address,
      onTap: () => editField("Address", "address",
          (value) => setState(() => address = value)),
    ),
    buildEditableField(
      icon: Icons.phone,
      label: "Phone",
      value: phone,
      onTap: () => editField(
          "Phone", "phone", (value) => setState(() => phone = value)),
    ),
    buildEditableField(
      icon: Icons.cake,
      label: "Birthday",
      value: birthday,
      onTap: () => editField("Birthday", "birthday",
          (value) => setState(() => birthday = value)),
    ),
    const SizedBox(height: 24),
    const Divider(),
    TextButton(
      onPressed: navigateToFavorites,
      style: TextButton.styleFrom(padding: const EdgeInsets.all(16)),
      child: const Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
```

```
                children: [
                  Text(
                    'My Favorites',
                    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
                  ),
                  Icon(Icons.favorite, color: Colors.red),
                ],
              ),
            ),
            TextButton(
              onPressed: navigateToCart,
              style: TextButton.styleFrom(padding: const EdgeInsets.all(16)),
              child: const Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  Text(
                    'My Cart',
                    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
                  ),
                  Icon(Icons.shopping_cart, color: Colors.green),
                ],
              ),
            ),
            const Spacer(),
            ElevatedButton(
              onPressed: () => signOut(context),
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.black,
                minimumSize: const Size(double.infinity, 50),
              ),
              child: const Text(
                'Sign Out',
                style: TextStyle(fontSize: 20, color: Colors.white),
              ),
            ),
          ],
        ),
      ),
    ),
  );
}

Widget buildEditableField({
  required IconData icon,
  required String label,
  String? value,
  required VoidCallback onTap,
}) {
  return GestureDetector(
    onTap: onTap,
    child: Padding(
      padding: const EdgeInsets.symmetric(vertical: 8.0),
      child: Row(
        children: [
          Icon(icon, size: 30),
          const SizedBox(width: 16),
          Text(
            value ?? label,
            style: TextStyle(
```

```
              fontSize: 16,
              color: value == null ? Colors.grey : Colors.black,
              fontWeight: value == null ? FontWeight.normal : FontWeight.bold,
            ),
          ),
        ],
      ),
    ),
  );
 }
}
```

Splash_screen.dart

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

class SplashScreen extends StatefulWidget {
 const SplashScreen({super.key});

 @override
 State<SplashScreen> createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
 @override
 void initState() {
  super.initState();
  _navigateToLogin();
 }

 void _navigateToLogin() async {
  await Future.delayed(const Duration(seconds: 3)); // Splash screen duration
  Navigator.pushReplacementNamed(context, '/login');
 }

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   backgroundColor: Colors.black,
   body: Container(
    height: MediaQuery.of(context).size.height,
    width: MediaQuery.of(context).size.width,
    decoration: const BoxDecoration(
     image: DecorationImage(image: AssetImage("assets/images/splash.png"),
     fit: BoxFit.cover)
    ),
   ),
  );
 }
}
```

app_style.dart

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

TextStyle appStyle(double size, Color color, FontWeight fontWeight) {
 return GoogleFonts.poppins(
```

```
    fontSize: size, color: color, fontWeight: fontWeight);
}

TextStyle appstyleWithHt(double size, Color color , FontWeight fw, double ht) {
  return GoogleFonts.poppins(fontSize: size, color: color, fontWeight: fw, height: ht);
}
```

bottom_nav_widget.dart

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

class BottomNavWidget extends StatelessWidget {
  const BottomNavWidget({
    super.key,
    this.onTap,
    this.icon,
  });
  final void Function()? onTap;
  final IconData? icon;

  @override
  Widget build(BuildContext context) {
    return GestureDetector(
      onTap: onTap,
      child: SizedBox(
        height: 40,
        width: 40,
        child: Icon(icon, color: Colors.white),
      ),
    );
  }
}
```

Bottom_nav.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_vector_icons/flutter_vector_icons.dart';
import 'package:provider/provider.dart';
import 'package:sneaker_app/controllers/mainscreen_provider.dart';
import 'bottom__nav_widget.dart';

class BottomNav extends StatelessWidget {
  const BottomNav({
    super.key,
  });

  @override
  Widget build(BuildContext context) {
    return Consumer<MainScreenNotifier>(
      builder: (context, mainScreenNotifier, child) {
        return SafeArea(
          child: Padding(
            padding: const EdgeInsets.all(8),
            child: Container(
              padding: const EdgeInsets.all(10),
              margin: const EdgeInsets.symmetric(horizontal: 10),
              decoration: const BoxDecoration(
                color: Colors.black,
                borderRadius: BorderRadius.all(Radius.circular(16)),
```

```
            ),
            child: Row(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: [
                BottomNavWidget(
                  onTap: () {
                    mainScreenNotifier.pageIndex = 0;
                  },
                  icon: mainScreenNotifier.pageIndex == 0
                      ? MaterialCommunityIcons.home
                      : MaterialCommunityIcons.home_outline,
                ),
                BottomNavWidget(
                  onTap: () {
                    mainScreenNotifier.pageIndex = 1;
                  },
                  icon: mainScreenNotifier.pageIndex == 1
                      ? Ionicons.heart
                      : Ionicons.heart_circle_outline,
                ),
                BottomNavWidget(
                  onTap: () {
                    mainScreenNotifier.pageIndex = 2;
                  },
                  icon: mainScreenNotifier.pageIndex == 2
                      ? Ionicons.cart
                      : Ionicons.cart_outline,
                ),
                BottomNavWidget(
                  onTap: () {
                    mainScreenNotifier.pageIndex = 3;
                  },
                  icon: mainScreenNotifier.pageIndex == 3
                      ? Ionicons.person
                      : Ionicons.person_outline,
                ),
              ],
            ),
          ),
        ),
      );
    },
  );
 }
}
```

Category_button.dart

```
import 'package:flutter/material.dart';
import 'package:sneaker_app/views/shared/app_style.dart';

class CategoryBtn extends StatelessWidget {
  const CategoryBtn(
      {super.key, this.onPress, required this.buttonClr, required this.label});
  final void Function()? onPress;
  final Color buttonClr;
  final String label;

  @override
```

```
 Widget build(BuildContext context) {
  return MaterialButton(
    onPressed: onPress,
    child: Container(
      padding: const EdgeInsets.symmetric(horizontal: 14),
      height: 46,
      decoration: BoxDecoration(
          border: Border.all(
            width: 1,
            color: buttonClr,
            style: BorderStyle.solid,
          ),
          borderRadius: const BorderRadius.all(Radius.circular(9))),
      child: Center(
        child: Text(
          label,
          style: appStyle(16, buttonClr, FontWeight.w600),
        ),
      ),
    ),
  );
 }
}
```

Checkout_btn.dart

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

class CheckoutButton extends StatelessWidget {
 const CheckoutButton({super.key, this.onTap, required this.label});

 final void Function()? onTap;
 final String label;

 @override
 Widget build(BuildContext context) {
  return GestureDetector(
    onTap: onTap,
    child: Padding(
      padding: const EdgeInsets.all(8),
      child: Container(
        decoration: const BoxDecoration(
            color: Colors.black,
            borderRadius: BorderRadius.all(Radius.circular(12))),
        height: 50,
        width: MediaQuery.of(context).size.width * 0.8,
        child: Center(
          child:
              Text(label, style: appStyle(18, Colors.white, FontWeight.w600)),
        ),
      ),
    ),
  );
 }
}
```

custom_spacer.dart

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

class CustomSpacer extends StatelessWidget {
  const CustomSpacer({super.key});

  @override
  Widget build(BuildContext context) {
    return const SizedBox(
      height: 25,
    );
  }
}
```

Home_widget.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_vector_icons/flutter_vector_icons.dart';
import 'package:provider/provider.dart';
import 'package:sneaker_app/controllers/product_provider.dart';
import 'package:sneaker_app/views/pages/product_by_cat.dart';
import 'package:sneaker_app/views/pages/product_page.dart';
import 'package:sneaker_app/views/shared/product_card.dart';
import '../../models/sneakers_model.dart';
import 'app_style.dart';
import 'new_shoes.dart';

class HomeWidget extends StatelessWidget {
  const HomeWidget({
    super.key,
    required Future<List<Sneakers>> male,
    required this.tabIndexNo,
  }) : _male = male;

  final Future<List<Sneakers>> _male;
  final int tabIndexNo;

  @override
  Widget build(BuildContext context) {
    var productNotifier = Provider.of<ProductNotifier>(context);

    return SingleChildScrollView(
      child: Column(
        children: [
          SizedBox(
            height: MediaQuery.of(context).size.height * 0.46,
            child: FutureBuilder<List<Sneakers>>(
              future: _male,
              builder: (context, snapshot) {
                if (snapshot.connectionState == ConnectionState.waiting) {
                  return const CircularProgressIndicator(
                    color: Colors.white,
                  );
                } else if (snapshot.hasError) {
                  return Center(child: Text("ERROR: $snapshot.error"),);
                } else {
                  final male = snapshot.data;
                  return ListView.builder(
                    itemCount: male!.length,
                    // vertical
```

```
                scrollDirection: Axis.horizontal,
                itemBuilder: (context, index) {
                  final shoe = snapshot.data![index];
                  return GestureDetector(
                    onTap: () {
                      productNotifier.shoeSizes = shoe.sizes;
                      Navigator.push(
                        context,
                        MaterialPageRoute(
                          builder: (context) => ProductPage(
                            id: shoe.id,
                            category: shoe.category,
                          ),
                        ),
                      );
                    },
                    child: ProductCard(
                        price: "\₹${shoe.price}",
                        category: shoe.category,
                        id: shoe.id,
                        name: shoe.name,
                        image: shoe.imageUrl[0]),
                  );
                },
              );
            }
          },
        ),
      ),
      Column(
        children: [
          Padding(
            padding: const EdgeInsets.fromLTRB(12, 20, 12, 20),
            child: Row(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: [
                Text(
                  "Latest Shoes",
                  style: appStyle(
                    24,
                    Colors.black,
                    FontWeight.bold,
                  ),
                ),
                GestureDetector(
                  onTap: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) => ProductByCategory(
                          tabIndex: tabIndexNo,
                        ),
                      ),
                    );
                  },
                  child: Row(
                    children: [
                      Text(
                        "Show All",
```

```
                    style: appStyle(
                      22,
                      Colors.black,
                      FontWeight.w500,
                    ),
                  ),
                  const Icon(
                    AntDesign.caretright,
                    size: 20,
                  )
                ],
              ),
            )
          ],
        ),
      ),
      SizedBox(
    height: MediaQuery.of(context).size.height * 0.14,
    child: FutureBuilder<List<Sneakers>>(
      future: _male,
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.waiting) {
          return const CircularProgressIndicator();
        } else if (snapshot.hasError) {
          return Text("ERROR: $snapshot.error");
        } else {
          final male = snapshot.data;
          return ListView.builder(
            // vertical
            itemCount: male!.length,
            scrollDirection: Axis.horizontal,
            itemBuilder: (context, index) {
              final shoe = snapshot.data![index];
              return Padding(
                padding: const EdgeInsets.all(8.0),
                child: NewShoes(imageUrl: shoe.imageUrl[1]),
              );
            },
          );
        }
      },
    ),
  ),
    ],
  ),
  // close

 ],
 ),
 );
 }
}
```

Latest_shoes.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_staggered_grid_view/flutter_staggered_grid_view.dart';
import 'package:sneaker_app/views/shared/stagger_tile.dart';
import '../../models/sneakers_model.dart';
```

```
class LatestShoes extends StatelessWidget {
  const LatestShoes({
    super.key,
    required Future<List<Sneakers>> male,
  }) : _male = male;

  final Future<List<Sneakers>> _male;

  @override
  Widget build(BuildContext context) {
    return FutureBuilder<List<Sneakers>>(
        future: _male,
        builder: (context, snapshot) {
          if (snapshot.connectionState == ConnectionState.waiting) {
            return const CircularProgressIndicator();
          } else if (snapshot.hasError) {
            return Text("Error ${snapshot.error}");
          } else {
            final male = snapshot.data;
            return MasonryGridView.count(
              padding: EdgeInsets.zero,
              crossAxisCount: 2,
              crossAxisSpacing: 20,
              mainAxisSpacing: 16,
              itemCount: male!.length,
              scrollDirection: Axis.vertical,
              itemBuilder: (context, index) {
                final shoe = male[index];
                return StaggerTile(
                  imageUrl: shoe.imageUrl[1],
                  name: shoe.name,
                  price: "\₹${shoe.price}",
                );
              },
            );

          }
        });
  }
}
```

New_shoes.dart

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

class NewShoes extends StatelessWidget {
  const NewShoes({
    super.key,
    required this.imageUrl,
  });

  final String imageUrl;

  @override
  Widget build(BuildContext context) {
    return Container(
      decoration: const BoxDecoration(
```

```
      color: Colors.white,
      borderRadius: BorderRadius.all(Radius.circular(16)),
      boxShadow: [
        BoxShadow(
          color: Colors.white,
          spreadRadius: 1,
          blurRadius: 0.8,
          offset: Offset(0, 1),
        ),
      ],
    ),
    height: MediaQuery.of(context).size.height * 0.12,
    width: MediaQuery.of(context).size.width * 0.28,
    child: CachedNetworkImage(
      imageUrl: imageUrl,
    ),
   );
  }
}
```

Product_card.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_vector_icons/flutter_vector_icons.dart';
import 'package:provider/provider.dart';
import 'package:sneaker_app/views/pages/favorites.dart';
import 'package:sneaker_app/views/shared/app_style.dart';
import '../../controllers/favorites_provider.dart';
import '../../models/constants.dart';

class ProductCard extends StatefulWidget {
  const ProductCard(
      {super.key,
      required this.price,
      required this.category,
      required this.id,
      required this.name,
      required this.image});

  final String price;
  final String category;
  final String id;
  final String name;
  final String image;

  @override
  State<ProductCard> createState() => _ProductCardState();
}

class _ProductCardState extends State<ProductCard> {
  // final _favBox = Hive.box("fav_box");

  @override
  Widget build(BuildContext context) {
    var favouritesNotifier =
        Provider.of<FavouritesNotifier>(context, listen: true);
    favouritesNotifier.getFavourites();

    return Padding(
```

```
padding: const EdgeInsets.fromLTRB(8, 0, 20, 0),
child: ClipRRect(
 borderRadius: const BorderRadius.all(Radius.circular(16)),
 child: Container(
  padding: const EdgeInsets.symmetric(horizontal: 8),
  height: MediaQuery.of(context).size.height,
  width: MediaQuery.of(context).size.width * 0.64,
  decoration: const BoxDecoration(boxShadow: [
   BoxShadow(
     color: Color.fromARGB(255, 243, 243, 243),
     spreadRadius: 1,
     blurRadius: 0.6,
     offset: Offset(1, 1))
  ]),
  child: Column(
   crossAxisAlignment: CrossAxisAlignment.start,
   children: [
    Stack(
     children: [
      Padding(
       padding: const EdgeInsets.only(left: 12, right: 12),
       child: Container(
        height: MediaQuery.of(context).size.height * 0.23,
        decoration: BoxDecoration(
          image: DecorationImage(
            image: NetworkImage(widget.image),
            fit: BoxFit.contain)),
       ),
      ),
      Positioned(
       right: 10,
       top: 10,
       child: GestureDetector(
        onTap: () async {
         if (favouritesNotifier.ids.contains(widget.id)) {
          Navigator.push(
           context,
           MaterialPageRoute(
             builder: (context) => const Favourites()),
          );
         } else {
          await favouritesNotifier.createFav({
           "id": widget.id,
           "name": widget.name,
           "category": widget.category,
           "price": widget.price,
           "imageUrl": widget.image,
          });
          setState(() {
           ids.add(widget
             .id); // Update the state to include the new favorite
          });
         }
        },
        child: favouritesNotifier.ids.contains(widget.id)
          ? const Icon(AntDesign.heart)
          : const Icon(AntDesign.hearto),
       ),
      ),
```

```
                ],
              ),
              Padding(
                padding: const EdgeInsets.only(left: 8),
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      widget.name,
                      style: appstyleWithHt(
                          36, Colors.black, FontWeight.bold, 1.1),
                    ),
                    const SizedBox(
                      height: 2,
                    ),
                    Text(
                      widget.category,
                      style:
                          appstyleWithHt(18, Colors.grey, FontWeight.bold, 1.5),
                    )
                  ],
                ),
              ),
              Padding(
                padding: const EdgeInsets.only(left: 8, right: 8),
                child: Text(
                  widget.price,
                  style: appStyle(30, Colors.black, FontWeight.w600),
                ),
              )
            ],
          ),
        ),
      ),
    );
  }
}
```

Stagger_tile.dart

```
import 'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:sneaker_app/views/shared/app_style.dart';

class StaggerTile extends StatefulWidget {
  const StaggerTile(
      {super.key,
      required this.imageUrl,
      required this.name,
      required this.price});

  final String imageUrl;
  final String name;
  final String price;

  @override
  State<StaggerTile> createState() => _StaggerTileState();
}
```

```
class _StaggerTileState extends State<StaggerTile> {
  @override
  Widget build(BuildContext context) {
    return Container(
      height: MediaQuery.of(context).size.height*0.3,
      decoration: const BoxDecoration(
        color: Colors.white,
        borderRadius: BorderRadius.all(Radius.circular(16)),
      ),
      child: Padding(
        padding: const EdgeInsets.all(8),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.spaceAround,
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Center(
              child: CachedNetworkImage(
                imageUrl: widget.imageUrl,
                fit: BoxFit.fill,
              ),
            ),
            SizedBox(
              height: 92,
              child: Column(
                mainAxisAlignment: MainAxisAlignment.spaceAround,
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text(
                    widget.name,
                    style: appstyleWithHt(22, Colors.black, FontWeight.w700, 1.2),
                  ),
                  Text(
                    widget.price,
                    style: appstyleWithHt(20, Colors.black, FontWeight.w500, 1),
                  )
                ],
              ),
            )
          ],
        ),
      ),
    );
  }
}
```

Firebase_options.dart

```
// File generated by FlutterFire CLI.
// ignore_for_file: type=lint
import 'package:firebase_core/firebase_core.dart' show FirebaseOptions;
import 'package:flutter/foundation.dart'
    show defaultTargetPlatform, kIsWeb, TargetPlatform;

/// Default [FirebaseOptions] for use with your Firebase apps.
///
/// Example:
/// ```dart
/// import 'firebase_options.dart';
/// // ...
```

```
/// await Firebase.initializeApp(
///   options: DefaultFirebaseOptions.currentPlatform,
/// );
/// ```
class DefaultFirebaseOptions {
  static FirebaseOptions get currentPlatform {
    if (kIsWeb) {
      return web;
    }
    switch (defaultTargetPlatform) {
      case TargetPlatform.android:
        return android;
      case TargetPlatform.iOS:
        return ios;
      case TargetPlatform.macOS:
        return macos;
      case TargetPlatform.windows:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for windows - '
          'you can reconfigure this by running the FlutterFire CLI again.',
        );
      case TargetPlatform.linux:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for linux - '
          'you can reconfigure this by running the FlutterFire CLI again.',
        );
      default:
        throw UnsupportedError(
          'DefaultFirebaseOptions are not supported for this platform.',
        );
    }
  }

  static const FirebaseOptions web = FirebaseOptions(
    apiKey: 'AIzaSyAWgz9ALr-t-fzgrzQijNR2Hfgkqf1Zv7U',
    appId: '1:31400773149:web:db681e19220e46ebdc28f4',
    messagingSenderId: '31400773149',
    projectId: 'sneakrush-cbda5',
    authDomain: 'sneakrush-cbda5.firebaseapp.com',
    storageBucket: 'sneakrush-cbda5.firebasestorage.app',
  );

  static const FirebaseOptions android = FirebaseOptions(
    apiKey: 'AIzaSyBbg1Tt3XF093CU8Pkef8SzV5bm_Afuhro',
    appId: '1:306503120192:android:fee8496de49856a46276a5',
    messagingSenderId: '306503120192',
    projectId: 'sneakrush-d06de',
    storageBucket: 'sneakrush-d06de.firebasestorage.app',
  );

  static const FirebaseOptions macos = FirebaseOptions(
    apiKey: 'AIzaSyDTjaSxLLRmMEtTQ8S5zdziRwqQJ2Edt-4',
    appId: '1:306503120192:ios:e07aef0f5e8091dc6276a5',
    messagingSenderId: '306503120192',
    projectId: 'sneakrush-d06de',
    storageBucket: 'sneakrush-d06de.firebasestorage.app',
    iosBundleId: 'com.example.sneakerApp',
  );
```

```
  static const FirebaseOptions ios = FirebaseOptions(
    apiKey: 'AIzaSyDTjaSxLLRmMEtTQ8S5zdziRwqQJ2Edt-4',
    appId: '1:306503120192:ios:e07aef0f5e8091dc6276a5',
    messagingSenderId: '306503120192',
    projectId: 'sneakrush-d06de',
    storageBucket: 'sneakrush-d06de.firebasestorage.app',
    iosBundleId: 'com.example.sneakerApp',
  );

}
```

helper.dart

```
import 'package:flutter/services.dart' as the_bundle;
import '../models/sneakers_model.dart';

// this class fetches data from the json file and return it to the app
class Helper {
// Male
  Future<List<Sneakers>> getMaleSneakers() async {
    final data =
        await the_bundle.rootBundle.loadString("assets/json/men_shoes.json");

    final maleList = sneakersFromJson(data);

    return maleList;
  }

// Female
  Future<List<Sneakers>> getFemaleSneakers() async {
    final data =
        await the_bundle.rootBundle.loadString("assets/json/women_shoes.json");

    final femaleList = sneakersFromJson(data);

    return femaleList;
  }

// Kids
  Future<List<Sneakers>> getKidsSneakers() async {
    final data =
        await the_bundle.rootBundle.loadString("assets/json/kids_shoes.json");

    final kidsList = sneakersFromJson(data);

    return kidsList;
  }

  // Single Male
  Future<Sneakers> getMaleSneakersById(String id) async {
    final data =
        await the_bundle.rootBundle.loadString("assets/json/men_shoes.json");

    final maleList = sneakersFromJson(data);

    final sneaker = maleList.firstWhere((sneaker) => sneaker.id == id);

    return sneaker;
  }
```

```
 // Single Female
 Future<Sneakers> getFemaleSneakersById(String id) async {
  final data =
     await the_bundle.rootBundle.loadString("assets/json/women_shoes.json");

  final maleList = sneakersFromJson(data);

  final sneaker = maleList.firstWhere((sneaker) => sneaker.id == id);

  return sneaker;
 }

 // Single Kids
 Future<Sneakers> getKidsSneakersById(String id) async {
  final data =
     await the_bundle.rootBundle.loadString("assets/json/kids_shoes.json");

  final maleList = sneakersFromJson(data);

  final sneaker = maleList.firstWhere((sneaker) => sneaker.id == id);

  return sneaker;
 }
}
```

Constants.dart

```
List<dynamic> favor = [];

List<dynamic> ids = [];
```

sneakers_model.dart

```
import 'dart:convert';

List<Sneakers> sneakersFromJson(String str) =>
   List<Sneakers>.from(json.decode(str).map((x) => Sneakers.fromJson(x)));

class Sneakers {
 Sneakers({
  required this.id,
  required this.name,
  required this.category,
  required this.imageUrl,
  required this.oldPrice,
  required this.sizes,
  required this.price,
  required this.description,
  required this.title,
 });

 final String id;
 final String name;
 final String category;
 final List<String> imageUrl;
 final String oldPrice;
 final List<dynamic> sizes;
 final String price;
```

```
  final String description;
  final String title;

  factory Sneakers.fromJson(Map<String, dynamic> json) => Sneakers(
        id: json["id"],
        name: json["name"],
        category: json["category"],
        imageUrl: List<String>.from(json["imageUrl"].map((x) => x)),
        oldPrice: json["oldPrice"],
        sizes: List<dynamic>.from(json["sizes"].map((x) => x)),
        price: json["price"],
        description: json["description"],
        title: json["title"],
      );
}
```

Cart_provider.dart

```
import 'dart:convert';

List<Sneakers> sneakersFromJson(String str) =>
    List<Sneakers>.from(json.decode(str).map((x) => Sneakers.fromJson(x)));

class Sneakers {
  Sneakers({
    required this.id,
    required this.name,
    required this.category,
    required this.imageUrl,
    required this.oldPrice,
    required this.sizes,
    required this.price,
    required this.description,
    required this.title,
  });

  final String id;
  final String name;
  final String category;
  final List<String> imageUrl;
  final String oldPrice;
  final List<dynamic> sizes;
  final String price;
  final String description;
  final String title;

  factory Sneakers.fromJson(Map<String, dynamic> json) => Sneakers(
        id: json["id"],
        name: json["name"],
        category: json["category"],
        imageUrl: List<String>.from(json["imageUrl"].map((x) => x)),
        oldPrice: json["oldPrice"],
        sizes: List<dynamic>.from(json["sizes"].map((x) => x)),
        price: json["price"],
        description: json["description"],
        title: json["title"],
      );
}
```

Favorites_provider.dart

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

class FavouritesNotifier extends ChangeNotifier {
  final _favBox = Hive.box("fav_box");

  List<dynamic> _ids = [];
  List<dynamic> _favorites = [];
  List<dynamic> _fav = [];

  List<dynamic> get ids => _ids;

  set ids(List<dynamic> newIds) {
    _ids = newIds;
    notifyListeners();
  }

  List<dynamic> get favorites => _favorites;

  set favorites(List<dynamic> newFav) {
    _favorites = newFav;
    notifyListeners();
  }

  List<dynamic> get fav => _fav;

  set fav(List<dynamic> newFav) {
    _fav = newFav;
    notifyListeners();
  }

  getFavourites() {
    final favData = _favBox.keys.map((key) {
      final item = _favBox.get(key);

      return {"key": key, "id": item["id"]};
    }).toList();

    _favorites = favData.toList();
    _ids = _favorites.map((item) => item["id"]).toList();
  }

  getAllData() {
    final favData = _favBox.keys.map((key) {
      final item = _favBox.get(key);
      return {
        "key": key,
        "id": item["id"],
        "name": item["name"],
        "category": item["category"],
        "price": item["price"],
        "imageUrl": item["imageUrl"],
      };
    }).toList();

    _fav = favData.reversed.toList();
  }
```

```
  Future<void> deleteFav(int key) async {
    await _favBox.delete(key);
  }

  Future<void> createFav(Map<String, dynamic> addFav) async {
    await _favBox.add(addFav);
  }
}
```

Mainscreen_provider.dart

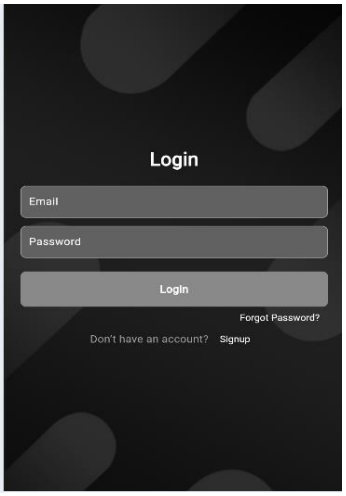
```
import 'package:flutter/material.dart';

class MainScreenNotifier extends ChangeNotifier {
  int _pageIndex = 0;

  int get pageIndex => _pageIndex;

  set pageIndex(int newIndex) {
    _pageIndex = newIndex;
    notifyListeners();
  }
}
```

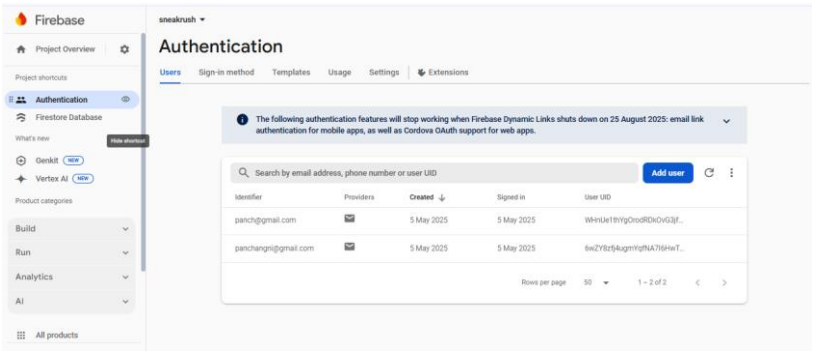
Product_provider.dart

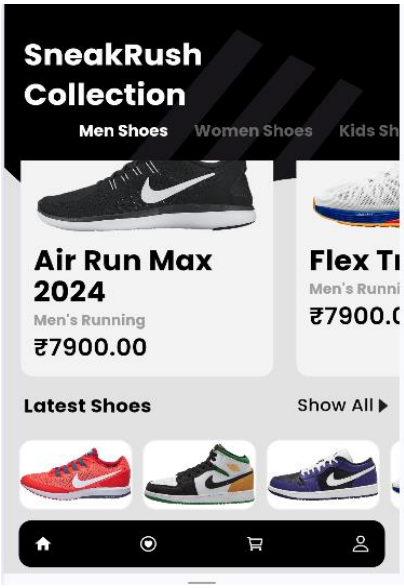
```
import 'package:flutter/material.dart';
import 'package:sneaker_app/models/sneakers_model.dart';
import '../services/helper.dart';

class ProductNotifier extends ChangeNotifier {
  int _activePage = 0;
  List<dynamic> _shoeSizes = [];
  List<String> _sizes = [];

  int get activePage => _activePage;

  set activePage(int newIndex) {
    _activePage = newIndex;
    notifyListeners();
  }

  List<dynamic> get shoeSizes => _shoeSizes;

  set shoeSizes(List<dynamic> newSizes) {
    _shoeSizes = newSizes;
    notifyListeners();
  }

  late Future<List<Sneakers>> male;
  late Future<List<Sneakers>> female;
  late Future<List<Sneakers>> kids;
  late Future<Sneakers> sneaker;

  void getMale() {
    male = Helper().getMaleSneakers();
  }
```

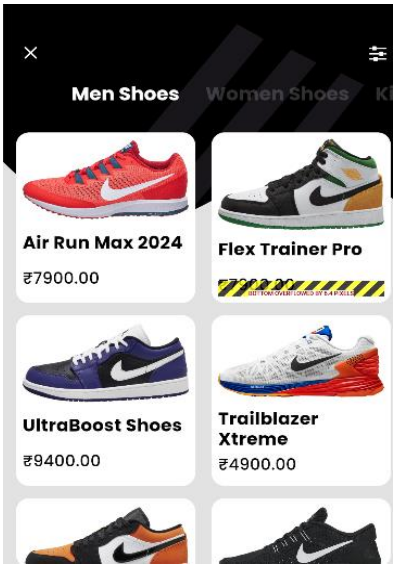
```
  void getFemale() {
    female = Helper().getFemaleSneakers();
  }

  void getkids() {
    kids = Helper().getKidsSneakers();
  }

  void getShoes(String category, String id) {
    if (category == "Men's Running") {
      sneaker = Helper().getMaleSneakersById(id);
    } else if (category == "Women's Running") {
      sneaker = Helper().getFemaleSneakersById(id);
    } else {
      sneaker = Helper().getKidsSneakersById(id);
    }
  }

//  function is called with an index, it will toggle the selection
//  of that item and leave the selection of all other items as they were.
//  This will allow for multiple items to be selected at once.

  void toggleCheck(int index) {
    for (int i = 0; i < _shoeSizes.length; i++) {
      if (i == index) {
        _shoeSizes[i]["isSelected"] = !_shoeSizes[i]["isSelected"];
      }
    }
    notifyListeners();
  }

  List<String> get sizes => _sizes;

  set sizes(List<String> newSizes) {
    _sizes = newSizes;
    notifyListeners();
  }
}
```

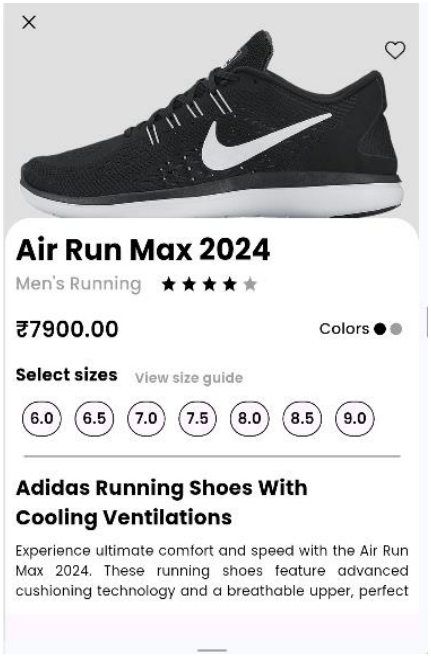
Screenshots

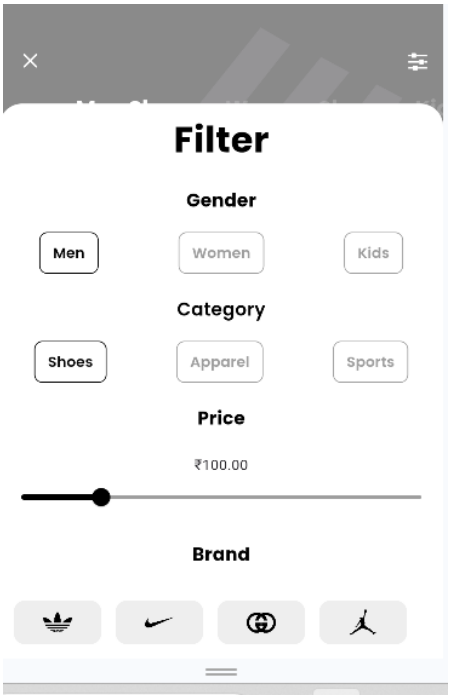



Firebase
Project Overview
Authentication
Firestore Database
Genkit
Vertex AI
Build
Run
Analytics
AI
All products
sneakrush
Authentication
Users
Sign-in method
Templates
Usage
Settings
Extensions
Add user

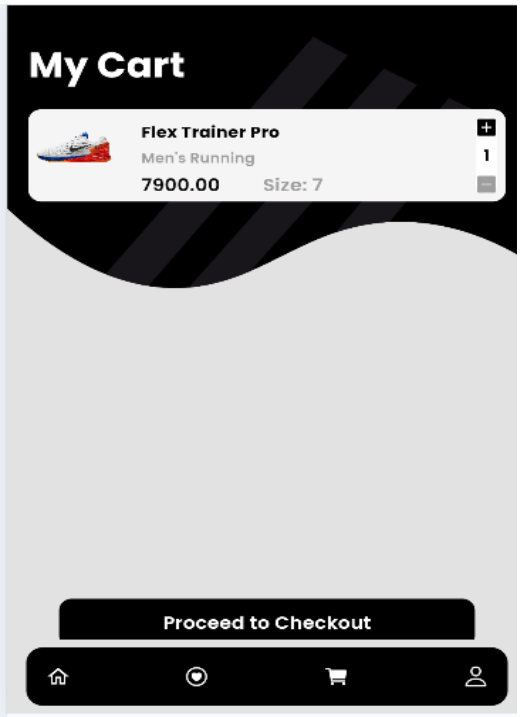

SneakRush
Collection
Men Shoes
Women Shoes
Air Run Max
2024
Men's Running
₹7900.00
Latest Shoes
Show All

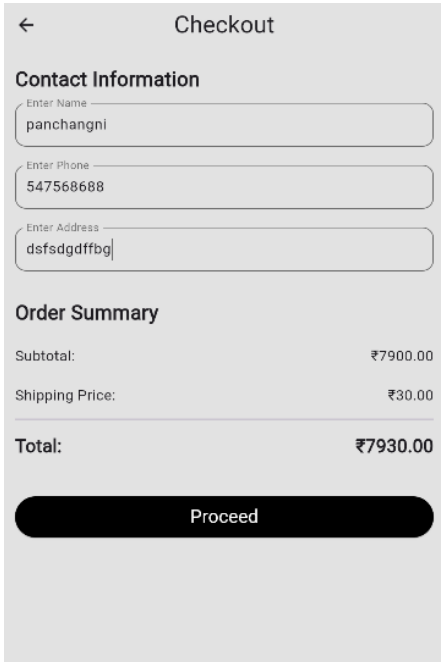

Men Shoes
Women Shoes
Air Run Max 2024
₹7900.00
Flex Trainer Pro
UltraBoost Shoes
₹9400.00
Trailblazer
Xtreme
₹4900.00

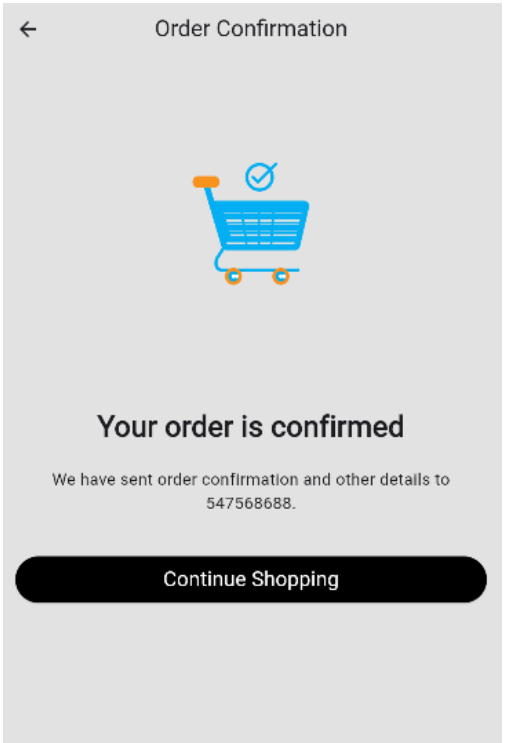

Air Run Max 2024
Men's Running
₹7900.00
Colors
Select sizes
View size guide
6.0
6.5
7.0
7.5
8.0
8.5
9.0
Adidas Running Shoes With Cooling Ventilations
Experience ultimate comfort and speed with the Air Run Max 2024. These running shoes feature advanced cushioning technology and a breathable upper, perfect

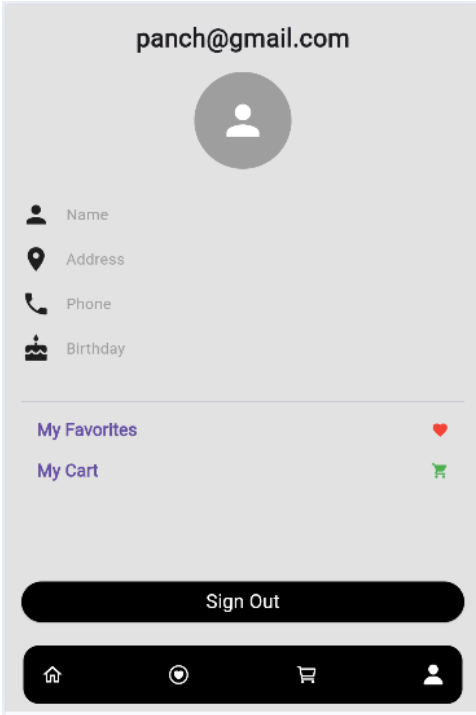

Filter
Gender
Men
Women
Kids
Category
Shoes
Apparel
Sports
Price
₹100.00
Brand

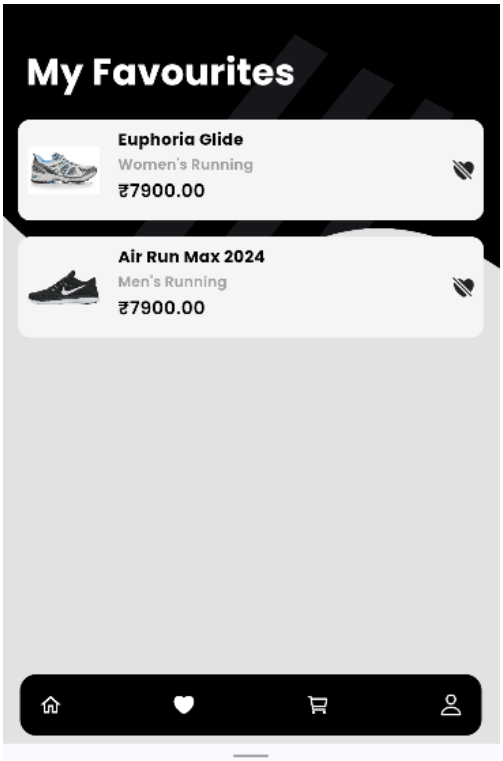
Select sizes
View size guide
6.0
6.5
7.0
7.5
8.0
8.5
9.0
Adidas Running Shoes With Cooling Ventilations
Stay agile and strong with the Flex Trainer Pro, designed for high-intensity workouts and training sessions. These training shoes offer superior stability and support with a lightweight, breathable upper. The high-grip sole ensures that you maintain your footing during intense movements, while the cushioned midsole keeps your
Add to Cart
Added to cart successfully!


My Cart
Flex Trainer Pro
Men's Running
7900.00
Size: 7
1
Proceed to Checkout


Checkout
Contact Information
Enter Name
panchangni
Enter Phone
547568688
Enter Address
dsfsdgdffbg
Order Summary
Subtotal:
₹7900.00
Shipping Price:
₹30.00
Total:
₹7930.00
Proceed


Order Confirmation
Your order is confirmed
We have sent order confirmation and other details to 547568688.
Continue Shopping


panch@gmail.com
Name
Address
Phone
Birthday
My Favorites
My Cart
Sign Out


My Favourites
Euphoria Glide
Women's Running
₹7900.00
Air Run Max 2024
Men's Running
₹7900.00

Conclusion	This Flutter project demonstrates the integration of Firebase for backend services and Hive for local data storage, making it a practical example of building a cross-platform eCommerce app. It uses the Provider package for state management, ensuring scalable and reactive UI updates across features like the main screen navigation, product listings, cart, and favorites. The app initializes Firebase differently for web and mobile using platform-aware logic, showing how to write adaptable code. Hive is used to persist cart and favorite items locally, enabling offline functionality. Additionally, flutter_screenutil ensures that the UI is responsive across different screen sizes. Navigation is handled through named routes, with a clear structure from splash to login/signup and main pages. Overall, the project covers essential concepts like authentication, state management, responsive design, and local data storage in a real-world app context.