Shopify Application

Name: Fatma Ahmed Hammad

In my application, I created screens using provider and firebase and quick alerts package.

I used types of providers to handle business logic and firebase to benefit its services.

After I connected my app with firebase, I got its packages.

In Splash Screen, I created its UI to show image in background and shopping cart icon and in it handle check user in function checkUser(), check if this is the user's first access to the application or user log out from app this mean user is null and after listen on it in FirebaseAuth if found it equal null from splash screen go to master screen that contains login screen, signup screen and forgot password screen too if he doesn't have an account? Go to sign up screen and create one, or if he already has one go to login screen to login to app. Or if he forgot password go to forgot password screen.

If user is founded from splash screen go to home screen directly without having to go to master screen. I call its function in initState() of this screen Because it is called immediately after entering the screen and its benefit to open stream to listen user directly.

In dispose() function, I cancel this stream because we didn't need to listen user after leaving splash screen.

In AuthProviderApp, I used ChangeNotifierProvider to handle business logic of login, signup, forgotpassword, toggleObscure and logout.

I assign one bool variable with true value and define three null variables

One GlobalKey<FormState>formKey to control form and two TextEditingController email and password to control two textformfield that take data from user.

I assign three variables with them values in init() function and reassign with null value in providerDispose() function because I will use them on more than one screen so, to avoid any error for ex: use form key in more than one place or overwrite text of textformfield for specific screen I get null value to them.

In toggleObscure() function we change boolean value of obscureText to Reverse the Currentvalue and add notifyListeners() to listen this change such as setstate in stateful

In logIn(), signUp() and resendEmail() , this three functions similar in some of parts such that check validates of current state of form if its true complete function Which is inside try catch to avoid any error and tell user when the operation completed successfully or there is an error and what is it by show message from QuickAlert package.

In login function, I use this function signInWithEmailAndPassword from instance of FirebaseAuth and pass text of emailcontroller and passwordcontroller to its arguments to available authentication from firebase for login to app If the user enters data identical to what was previously stored for him.

In signup function, I use this function createUserWithEmailAndPassword from instance of FirebaseAuth and pass text of emailcontroller and passwordcontroller to its arguments to available authentication from firebase for sign up and it stores the data entered by the user and is specific to him only.

In resend email function, I use this function sendPasswordResetEmail from instance of FirebaseAuth and pass text of emailcontroller its argument to available authentication from firebase for resend email and reset password to make user change it to new password in case he forgot it and help him to login again to app easily.

In sign out function, I use this function signOut from instance of FirebaseAuth to available authentication from firebase for sign out from application if user wanted to log out app.

So, go to splash screen again, so there are functions that can find them in this provider that used it in three screens in master screen.

In Master Screen, I used DefaultTabController with length = 3 for three screens and make tab bar has texts with screen names and show them in TabBarView widget.

In Log In Screen, Sign Up Screen and Forgot Password Screen, this three screens similar in some of parts such that there are StatefulWidget and I call init() function from AuthProviderApp in each initState() function related to the screen I choose it to reassign values to three variables special to this screen.

For UI to make it more reusable I extract widget to make suitable for every text form field in three screens and pass different data from one to another such text, icon, controller and validator that in email must be actual email ex: gmail and when password length must be 8 or more and in icon wrap with inkwell widget to call toggleObscure function from provider to make it available to show and hide password and every button in screen was extracted widget too to make suitable for every function special to FirebaseAuth that I call it in this screen to make them available I used Consumer widget that It allows obtaining values from AuthProviderApp that I gave a special word for it is value that use it for any call from this provider all extracted widgets I used inside in form widget such as email controller, pass controller and obscureText.

In log in screen, form widget contains two CustomTxtFieldWidget() one for email and other for password and CustomButtonWidget() with text log in and in on tap function call logIn() function from AuthProviderApp by value keyword.

In sign up screen, form widget contains two CustomTxtFieldWidget() one for email and other for password and CustomButtonWidget() with text sign up and in on tap function call signUp() function from AuthProviderApp by value keyword.

In forgot password screen, form widget contains one CustomTxtFieldWidget() one for email and CustomButtonWidget() with text send email and in on tap function call resendEmail() function from AuthProviderApp by value keyword.

For make my code more reusable I make some parts of the code that are constant in one dart file its name Constants to make it easy to call it by name and not repeat every times.

For Home Screen, after user log in application first screen he watched is home screen that contains in appbar icon named *shopping_cart* with badge that appears quantity of items in cart belonging to user, in body screens are displayed based on changing icons in the BottomNavigationBar when you click on them, from that home screen should be StatefulWidget to appear changes by setState() we used icons in BottomNavigationBar meaning to each screen we appear when pressed on it and change color of it to kPrimaryColor of app, we used onItemTap to make available change screens from list named widgetOptions that showed in body to screen that compatible with currentIndex, in BottomNavigationBar, first icon is home that meaning that screen containing details this screen named PartHomeScreen is StatelessWidget because it deals with more provider to every part in it, first part that known Categories, in this part we showed some of categories that be in app, this part deals with CategoriesProvider that such as serviceHolder can contains functions that we can call them from it in screen, before creating this provider, we create model for categories named CategoriesModel that contains all details special to every category such as id that unique, title, imagePath, backColor1, backColor2, imagePath and shadowColor and in provider we create list from this model to store every category we added in FirebaseFirestore by using getCategories function that be future function because it fetches data from internet and this function receive nullable limit parameter to make available limit or not number of categories that showed, in this function if limit == null that mean show all categories else showed some of categories that I passed it also we used to try catch to show error if it happened in try in this function we store in catsSnapshot its type is QuerySnapshot<Map<String, dynamic>> because we get data from collection in instance of FirebaseFirestore named categories order by field named createdAt after returning them we stored in List<CategoriesModel> that created before to deal with this provider in PartHomeScreen we used Consumer to listen everything and change that happened in provider to show this data and changes in UI

of screen because this provider is such provider as place holder for function, I used FutureBuilder because listens to the state of a Future, we used this widget instead of StreamBuilder because in this case data that is retrieved once as snapshot that FutureBuilder is used for displaying it while StreamBuilder is used for displaying data that is continuously updating, and listens to changes in a stream of data so in this part I used FutureBuilder for attribute named future I passed from value that indicates consumer that listen provider this function getCategories(context, limit: 3), and in attribute builder I checked state of snapshot if is loading or error happened or has data, and, in this case, I can show data in snapshot for this part of categories I passed this data for custom widget named RowCategoriesWidget in this widget I create row contains that list of categories was passed to this widget that every item in it was passed to other widget named CategoriesCircleWidget to show all data was stored, in this widget I created variable category its type of CategoriesModel that available all details in this model and VoidCallback onTap to available action when I click on each category, I showed every detail of category in column contains container contains image of category and color of background and shadow then I showed named of category then back to RowCategoriesWidget I added same CategoriesCircleWidget with special details to icon arrow with action when I click on it, I go to AllCategoriesScreen that showed
for user all categories in GridView that its data come from snapShot of FutureBuilder its attribute future: getCategories(context, limit: null), by this all categories was appeared in this screen by consumer that listen on CategoriesProvider, for GridView.builder its attributes itemCount is length of data of snapShot, crossAxisCount: 2 so showed two categories in every row in this screen and itemBuilder return CategoriesCircleWidget
for its category I passed data of snapShot with its index special with every category and in onTap by click on each category I go to the screen named AllProductsScreen and its attribute categoryId that I passed id of category that I click on it so make available to show

all products that same this category in this screen, I notice that many screens contains same appbar so create custom appbar for them named CustomAppBarWidget that returns AppBarWidget with action attribute icon close when clicking on it go to previous screen, for attribute appBar of every scaffold type is PreferredSizeWidget? So, we should add Size get preferredSize => const Size.fromHeight(50); to be available for all screens I used it in them, to benefit from Firebase services I stored all images that I used them in application in Storage in firebase to make available to change it in time I need to do it without back to code like we do when store in assets file.

In second part of PartHomeScreen I showed latest of advertisement for this part I created model named AdvertisementsModel contains id, imageURL and title and make provider like CategoriesProvider named AdsProvider, in it I create list of type AdvertisementsModel and I stored in it what return from function named getAdvertisements its type of future of list of AdvertisementsModel that contains all data of documents in collection named ads in firestore in firebase and passed them to variables in model and every id we get that can id make by default by firebase with can be unique for every document, go back to screen we use to consumer that listen to AdsProvider and in its attribute builder return FutureBuilder and in its attribute future passed from value of consumer access getAdvertisements function, for its attribute builder I checked states of snapshot if it has data, I return CarouselSliderWidget to its attribute I passed data of snapshot, in this widget we used to access data list of AdvertisementsModel that data stored before in it, to show data we used CarouselSlider to show data as advertisements its attribute items I passed items in list that I created before map for each item and show in container contains image as background of container with fill it and title as child of it and in attribute options we make autoPlay: true that means CarouselSlider moves on its own.

In third part of PartHomeScreen I showed some products were in app for this part I create model named ProductsModel that contains id, name, price, image, list of colors, list of sizes and categoryId then make provider like CategoriesProvider named ProductsProvider in it, I created list of ProductsModel that stored what return from function named getProducts with parameter nullable limit this type of function is future of list of ProductsModel in try in this function we store in productsSnapshot its type is QuerySnapshot<Map<String, dynamic>> because we get data from collection that contains more documents in instance of FirebaseFirestore named products order by field named createdAt after returning them we stored in List< ProductsModel > that created before that id of product that be default id that firebase give it to each product and categoryId that be default id that firebase give it to each before in categories collection and I create other function named getProductsById with parameter productId to get product by its id in this function we store in productsSnapshot its type is DocumentSnapshot<Map<String, dynamic>> because we get data from collection that contains more documents in instance of FirebaseFirestore named products but we specified one document we need by productId if productsSnapshot existes we return product with all details of it else return null, go back to screen we use to consumer that listen to ProductsProvider and in its attribute builder return FutureBuilder and in its attribute future passed from value of consumer access getProducts(context,limit: 3) function to show three of products, for its attribute builder I checked states of snapshot if it has data, I return FlexibleGridView to its attribute axisCount I show three products in row, in its attribute children I passed data of snapshot, map for item in data I passed to attribute product in ProductWidget, and other attribute onTap make available when clicking on this card I will go to ProductDetailsScreen and passed that item I click on it to its attribute product, in ProductWidget we used to access data of AdvertisementsModel variable of same type and onTap function to available clicking on it, to show data we used container includes column contains image, name and price of each product, in AllProductsScreen we found the same code of third part of PartHomeScreen but with some differences that in this screen, I create variable categoryId to receive id of each category I click on it and if snapshot has data I created filteredProducts variable has all data of snapshot that categoryId in its element == categoryId that passed to this screen then converts to list and check this list is not empty pass this list to children to FlexibleGridView and complete steps as in third part of PartHomeScreen. When I go to ProductDetailsScreen its type of StatefulWidget I create product variable its type of ProductsModel, Before build this screen we should go to create cart provider because I used many functions from this provider to this screen, so I create model for cart named CartModel in this I create nullable list to items that can be in cart its type of class named CartItem I created in it nullable variables for details of each item in cart such as itemId that unique id for each item, productId that id of product from which this item is made, quantity, selectColor and valueSize were selected in ProductDetailsScreen if user didn't select color or size the default keyword was stored in any of them was Nothing after finishing this class I stored all details I get from it to list that I created before in model of cart if this list not equal null, go to CartModel, in it I created variable named cartItem of type nullable CartItem and create function named createItemInstance to create instance of CartItem class and assigned it to cartItem variable and create getter named cartStream of type Stream<DocumentSnapshot<Map<String, dynamic>>> this stream listens to document which be email of user that was login app with it and we can get it from FirebaseAuth that stored data of user, so in collection cart in FirebaseFirestore I can get data from each docs by opening this stream, we can do without notify listener so this provider such as place holder, then I created list named products of type ProductsModel that I could store all products that user chosen in it and create function named onAddProductToList with two parameters product type of ProductsModel and cart type of CartModel in this function check index by if id of element in list == id of product that pass as parameter if true that mean is this product existed in list or if index == -1 that mean not found and add to list

and variable named total to store total price in it type double with initial value 0 and variable named totalNotifier of ValueNotifier with double type to listen for change of total variable every time and create calculateTotal function in it make loop to walk on every item in cart if products list was empty exit from loop else continue and in product variable check if id of every element in list products == productId of item and add this keyword firstOrNull to this check to avoid this error bad state no element if list was empty and if this variable not equal null calculates total of price by add value of total to result of multiplication of price of product with quantity of item to show value of total I passed it to value of totalNotifier and showed value in new frame by function addPostFrameCallback
to add item to cart I create onAddItemToCart function in try of it make Boolean variable isEqual with false value and nullable variable updatesItemId and in result variable stored data get from collection cart with document email of user and check if result exists in it created fireBaseCartItems that stored all items of cart then make loop to walk on every item in this variable and check if productId, selectColor ?? nothing and valueSize??nothing of item in cart == productId, selectColor ?? nothing and valueSize??nothing of item I added if all condition true this isEqual = true, the same item, give value of updatedItemId = itemId of item if two value is assigned to their variable this same item and wanted to increase its quantity instead of adding the same item with same details but we can't update value in list directly so I stored data of item I want to update in updatedItem variable and remove old item from cart and updated quantity to newest value to updatedItem variable then add updatedItem to list of items and update all items in cart with data in fireBaseCartItems else if items not same update items by union value of field with data from cartItem or if user first added to items in cart so he set data from cartItem. Other functions in this provider share with this function for more parts of code so for not repeating I tell differences in functions
to remove item from cart, we can update not remove array 'items' so this what I make in onRemoveItemFromCart that has string itemId, CartModel cart as parameters in try of this

function I confirm user to ensure that he wanted to remove this item or not so if result was true I search in items in cart about item that has itemId == itemId I passed to this function and remove it then update all items in cart and update total price by calling this function calculateTotal(cart) after remove item. To increase quantity of item I created function onIncreaseItemQuantity that has string itemId, CartModel cart as parameters in try of this I stored item I wanted in updatedItem variable and in index variable I stored index of its item then remove old item and calculate quantity of updatedItem that = quantity of old item +1 then add new item to items in cart but not use add() function, use insert() function because that take two parameters first index that position I wanted to add item in it and second item I used this function because I wanted to improve UI of cart screen, and prevent move cards in it ups and downs every time user increases or decreases item because add item in list by default be at the end of list after that update cart and update total price after increase quantity of item. In onDecreaseItemQuantity I used same code of previous function with little difference I checked quantity of updatedItem if it == 1 that mean when user decrease it should be removed so I call onRemoveItemFromCart and return (this point to give order to exit from code after do function) that I created before else calculate quantity of updatedItem that = quantity of old item – 1 and update cart and total price.

Go back to details screen in initState of it I call createItemInstance from CartProvider to every time open this screen create instance of cartItem was empty but adding was in firebase, in appbar of this screen I showed name and price for product that I clicked on it and I can access these values by product type ProductModel I created before, in part actions of appbar I put shopping cart with badge has number of quantity in cart I can found this number by making this badge child of StreamBuilder that I told its benefits and differences of FutureBuilder before, by this we can open stream to show every change of quantity for this attribute stream I called cartStream function from CartProvider and in attribute builder I checked snapshot if has data if true in condition assign initial value 0 to quantity and make loop for every item in items in cart deal item as Map<String, dynamic> not as object in snapshot.data?.data()?['items'] that first data meaning to document , second data meaning to what searching in items list and calculate quantity = old value of quantity + number of quantities in every item in items list then show final number of quantities in text of badge, in body of details screen I showed image of product and listview.builder axis.horizontal of list of colors that I stored in firestore with product make it available when I clicked one item of them appear check sign meaning to this color is chosen by using selectedColorCallBack as call back function and I make if condition in it to assign for every color its name then I pass this value to selectColor of cartItem from CartProvider to save this value to item I choose, in list sizes I take same steps of list colors without if condition to assign name of value and change was in color of size number to red then I pass this value to valueSize of cartItem from CartProvider to save this value to item I choose, and when I click on ADD TO CART button there some statements were done first, I passed id of product to productId of cartItem with CartProvider, second, I passed 1 to quantity of cartItem with CartProvider, third, I used uuid to generate random id and not repeat anymore and passed its value to itemId of cartItem with CartProvider, finally I called onAddItemToCart with CartProvider to add item to list of items in cart.

In cart screen, I used CustomAppBar in appbar and in body I used Streamuilder that I told before because with it I can open stream and listen every change was happened in its attribute stream I passed cartStream function from CartProvider and in its attribute builder I checked states of snapshot if snapshot has data, I stored in data variable all data in all documents in snapshot and if items in data variable not empty and I showed all items in ListView.builder not ListView because ListView creating all items at once unlike ListView.builder creates items as they're scrolled onto the screen and This approach is more memory-efficient and its child was FutureBuilder to show data came from future and toke snapshot of it for its attribute future I passed getProductsById function from

CartProvider and passed to productId parameter productId of item in index of data list and in attribute builder I checked data in snapshot if not equal null I called getProductsById I passed data of snapshot as ProductModel and data variable that I created before to store data that came from stream that listen to cart as CartModel and I called calculateTotal function and passed this data variable to function note this function without listen: false because I needed to listen every change in total price that this function calculates total and I passed all data to custom widget named CartItemWidget in imagePath take image of data of snapshot and in nameTxt take name of data of snapshot and in priceTxt take price of data of snapshot that three elements their data come from ProductModel in contextDelete take screenContext (send context of screen to recoginze context for loading in remove function) in cartData take data variable and in quantity take quantity of item in current index in data and in itemId take itemId of item in current index in data and in typeTxt take in color selectColor of item in current index in data and in size valueSize of item in current index in data and when enter in this custom widget every card would appear contains image, name and price of product and quantity, color and size of item and if user need increase or decrease quantity there are two icons behind the number of quantity when clicking on add icon that called onIncreaseItemQuantity function and do steps of code I told before in CartProvider and result increase quantity and update total price at the same when he click on other icon to decrease quantity that called onDecreaseItemQuantity function and do steps of code I told before in CartProvider and if he wanted to remove item from cart, he click on close icon that called onRemoveItemFromCart that result was remove item and update total price and go back to CartScreen to show total price I used ValueListenableBuilder that because some reasons: 1- using ValueListenableBuilder to listen total price in every change, 2- setState these changes (new value) without using ChangeNotifierProvider, 3- usable from listener because rebuild (refresh) for such this value not all screen, in its attribute valueListenable I passed totalNotifier from CartProvider and in its attribute builder I passed value that come from totalNotifier every time change in ProfileScreen, I used same strategy of CartScreen, I created model named ProfileModel for data that can be special for every profile that was created by every user and created ProfileProvider for business logic in it I created profileData variable its type ProfileModel then created createProfileInstance function to take instance of ProfileModel and created getter named profileStream its type of Stream its type <DocumentSnapshot< Map<String, dynamic>>> to open stream listen to every change in data that stored in Firestore in document named with email of user and collection named profiles and I created saveDataToProfile function to save any data user wanted to save if he entered this data for first time or he updated its data that stored before, in this function I created result variable to get doc (user email) of collection than in existingData variable I store all data that was I can reach to it from result variable if it was null stored empty map and I checked all three elements that can I access them by profileData if any one of them could not be empty I passed its value to value of key with same name of this element in existingData but if could be empty I passed value of key with same name of this element of data in result to value of key with same name of this element in existingData to if user don't enter new data I showed the old data he stored to him so if result exists, I passed existingData to update function to update all data in firestore but if not, I passed existingData to set function to add data to firestore for first time and created getDataFromProfile function its type of future of ProfileModel that maybe null in it I created result variable its type that maybe null DocumentSnapshot<Map<String, dynamic>>? Because we get data from document if this document exists I return data from it that this function used to display this data in ProfileScreen, In EditProfileScreen its StatefullWidget in initState function I called createProfileInstance to every time open this screen create instance of Profile was empty but adding was in firebase than I created two TextEditingController one for name and one for phone and variable imageUrl its type string I put icon of image and wrap it by GestureDetector to make it clickable, so to make user able to choose image I used

ImagePicker to pick image from gallery of user phone and I checked if null exit this function If not continue in filename variable I sorted name of image that I make by microsecond of TimeDate of now to make each image selected take microsecond of the current time and date and get the reference to storage root and I created the image folder first with name profiles and insider folder we upload the image and created reference for the image to be stored and I uploaded image (file) with type 'image/png' to stored successfully in storage then I get its URL and passed to imageUrl variable, and I used CustomTxtFieldWidget for two text field to take data that user entered to name and phone then I take CustomButtonWidget to button I saved data with it and in its function, I passed imageUrl variable to imageUrl of profileData from ProfileProvider and I passed text of nameController variable to name of profileData from ProfileProvider and I passed text of phoneController variable to phone of profileData from ProfileProvider and I called saveDataToProfile function from ProfileProvider to save all data user entered and in ProfileScreen I used StreamBuilder that I told its benefit before to listen every change in data of doc of collection and in its attribute steam I passed profileStream of ProfileProvider and in its attribute builder I checked snapshot and its states if snapshot has data, I used FutureBuilder to show data from future and in its attribute future I passed function getDataFromProfile from ProfileProvider then I showed data for image I used image.network and passed image of data of snapshot (URL of that stored in firebase storage) and I used two Text one I passed name of data of snapshot other I passed phone of data of snapshot also I showed email of user that by email of current user of instance from FirebaseAuth and created two button one for edit profile when I clicked on it, I go to EditProfileScreen one for logout when I clicked on it called signOut function from AuthProviderApp that I told before with-it user can log out application, in class named NotificationService I used for contains all function that can handle sending notification to application and message for each state for this I used FirebaseMessaging and created fcm variable its type of it to take token of mobile and check states of app (foreground, background, background or killed) and two Boolean variables with initial value false isInitalised refer to that user login to application other isTokenInit refer to that token of user and integer variable tries with initial value 0 and create init() function its type future that called in SplashScreen because when open app I check if user auth get token and store in firestore and in this function I checked if isInitalised was true I called *sendToken*() to send token and exit from function else continue in fcm variable stored instance of FirebaseMessaging and call function from instance to handle how to present notification setForegroundNotificationPresentationOptions with make sound and show badge and alert from FirebaseMessaging I call *onMessage* to open stream to listen message and in it I called *handleOnNotificationReceived* and passed this message and optional parameter isForeground true to make it handle notification when app open in Foreground and from FirebaseMessaging I call *onMessageOpenedApp* to open stream to listen message and in it I called *handleOnNotificationReceived* and passed this message parameter to make it handle notification when app open in Background Not killed and from FirebaseMessaging I call *onBackgroundMessage* and passed *firebaseMessagingBackgroundHandler* as handler in try I called *sendToken* and give true value to isInitalised and in catch give true value to isInitalised and checked number of tries if <= 20 I called init every time and increase tries to ensure that token send, and all function was done and create *onPushNotificationClosed* in it I give false value to isTokenInit and this function called in signOut function because when user logout then login that firebase maybe don't make token Although he is new user then create *sendToken* to send userToken to firestore that token only toke for auth person in it I checked isTokenInit was true mean that user token toke so exits from function else continue in userToken I stored token of user that I get it from fcm and this userToken stores as value of key token in map set as data in document named with email of user in collection named tokens in firestore then give true value to isTokenInit and create function *firebaseMessagingBackgroundHandler* to handle message for current platform (android, ios……) that function as like that used to initialize app with current platform that used in main function after connected app with firebase and firebase options file with firebase core I created *handleOnNotificationReceived* that trigger when foreground, background or killed when notification received and can handle background or killed but as foreground, I handle what happened so it has optional parameter with initial value false, true if is foreground so I used OverlaySupport widget to appear notification with showSimpleNotification I show title and body of notification to user to make easy to use it in any position in app should wrap MaterialApp with OverlaySupport.global that same all providers that I used to make it available to access any this in provider in any position of code in runApp function I wrap MyApp with MultiProvider and in list of it I put all providers that I used in code of my app I used jsonDecode for value of payload of data of message because payload come as string So, I used it to Parses the string and returns the resulting Json object and I created function *handleOnNotificationClicked* to call when I click on notification It will only be achieved when foreground and what make is in is Foreground but if background or killed open app by default and created *checkNotificationOnKilledApp* I called it in SplashScreen and when open app from notification. In CartScreen after user finish all changes that he wanted and wanted to place order he click on button named place order that take him to screen named OrderPlacedScreen to tell him that his order placed successfully and if he wanted to see all his orders click on button named my orders that take him to screen MyOrdersScreen, in this screen I used strategy that I used in CartScreen typical and all steps to show all details for all orders because everything was changed in cart screen meaning that changes in orders. Sorry, I forgot to say that I checked quantity of product in onAddProductToList() function because if user ordered quantity of item of this product decrease this quantity from quantity of product to if quantity of product == 0 this meaning to this product is out of stock and can’t be added to the list of products.