FOODEX-AN ONLINE FOOD ORDERING APP

# MainActivity.kt

package com.example.foodorderapp  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.layout.fillMaxSize  
import androidx.compose.material3.MaterialTheme  
import androidx.compose.material3.Surface  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Modifier  
import androidx.lifecycle.viewmodel.compose.viewModel  
import com.example.foodorderapp.navigation.NavGraph  
import com.example.foodorderapp.ui.theme.FoodOrderAppTheme  
import com.example.foodorderapp.viewmodel.CartViewModel  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *setContent* **{** FoodOrderApp()  
 **}** }  
}  
  
@Composable  
fun FoodOrderApp() {  
 FoodOrderAppTheme **{** Surface(  
 modifier = Modifier.*fillMaxSize*(),  
 color = MaterialTheme.colorScheme.background  
 ) **{** val cartViewModel: CartViewModel = viewModel()  
 NavGraph(cartViewModel = cartViewModel)  
 **}  
 }**}

# Cartitem.kt

package com.example.foodorderapp.model  
  
data class CartItem(  
 val foodItem: FoodItem,  
 val quantity: Int  
) {  
 val totalPrice: Double  
 get() = foodItem.price \* quantity  
}

# FoodItem.kt

package com.example.foodorderapp.model  
  
data class FoodItem(  
 val id: String,  
 val name: String,  
 val description: String,  
 val price: Double,  
 val imageUrl: String,  
 val restaurantId: String,  
 val category: String  
)

# Restaurant.kt

package com.example.foodorderapp.model  
  
data class Restaurant(  
 val id: String,  
 val name: String,  
 val description: String,  
 val imageUrl: String,  
 val rating: Float,  
 val deliveryTime: String,  
 val deliveryFee: String,  
 val minOrder: String  
)

# NavGraph.kt

package com.example.foodorderapp.navigation  
  
import androidx.compose.runtime.Composable  
import androidx.compose.runtime.getValue  
import androidx.lifecycle.compose.collectAsStateWithLifecycle  
import androidx.navigation.NavType  
import androidx.navigation.compose.NavHost  
import androidx.navigation.compose.composable  
import androidx.navigation.compose.rememberNavController  
import androidx.navigation.navArgument  
import com.example.foodorderapp.model.FoodItem  
import com.example.foodorderapp.model.Restaurant  
import com.example.foodorderapp.ui.screens.CartScreen  
import com.example.foodorderapp.ui.screens.RestaurantDetailScreen  
import com.example.foodorderapp.ui.screens.RestaurantListScreen  
import com.example.foodorderapp.viewmodel.CartViewModel  
  
sealed class Screen(val route: String) {  
 object RestaurantList : Screen("restaurantList")  
 object RestaurantDetail : Screen("restaurantDetail/{restaurantId}") {  
 fun createRoute(restaurantId: String) = "restaurantDetail/$restaurantId"  
 }  
 object Cart : Screen("cart")  
}  
  
@Composable  
fun NavGraph(cartViewModel: CartViewModel) {  
 val navController = rememberNavController()  
 val cartItems by cartViewModel.cartItems.collectAsStateWithLifecycle()  
  
 // Sample data  
 val restaurants = *listOf*(  
 Restaurant(  
 id = "1",  
 name = "Burger King",  
 description = "Fast food restaurant",  
 imageUrl = "https://example.com/burger-king.jpg",  
 rating = 4.5f,  
 deliveryTime = "20-30 min",  
 deliveryFee = "$2.99",  
 minOrder = "$10.00"  
 ),  
 Restaurant(  
 id = "2",  
 name = "Pizza Hut",  
 description = "Pizza restaurant",  
 imageUrl = "https://example.com/pizza-hut.jpg",  
 rating = 4.2f,  
 deliveryTime = "25-35 min",  
 deliveryFee = "$1.99",  
 minOrder = "$15.00"  
 )  
 )  
 val foodItems = *listOf*(  
 FoodItem(  
 id = "f1",  
 name = "Whopper",  
 description = "Signature flame-grilled beef burger",  
 price = 320.0,  
 imageUrl = "https://example.com/whopper.jpg",  
 restaurantId = "1",  
 category = "Burger"  
 ),  
 FoodItem(  
 id = "f2",  
 name = "Chicken Fries",  
 description = "Crispy chicken fries",  
 price = 300.0,  
 imageUrl = "https://example.com/chicken-fries.jpg",  
 restaurantId = "1",  
 category = "Sides"  
 ),  
 FoodItem(  
 id = "f3",  
 name = "Pepperoni Pizza",  
 description = "Classic pepperoni pizza",  
 price = 450.0,  
 imageUrl = "https://example.com/pepperoni-pizza.jpg",  
 restaurantId = "2",  
 category = "Pizza"  
 ),  
 FoodItem(  
 id = "f4",  
 name = "Garlic Bread",  
 description = "Freshly baked garlic bread",  
 price = 300.0,  
 imageUrl = "https://example.com/garlic-bread.jpg",  
 restaurantId = "2",  
 category = "Sides"  
 ),  
 FoodItem(  
 id = "f5",  
 name = "Paneer Tikka Pizza",  
 description = "Spicy paneer tikka on a cheesy pizza base",  
 price = 480.0,  
 imageUrl = "https://example.com/paneer-tikka-pizza.jpg",  
 restaurantId = "2",  
 category = "Pizza"  
 ),  
 FoodItem(  
 id = "f6",  
 name = "Veggie Supreme Burger",  
 description = "Loaded with fresh veggies and cheese",  
 price = 350.0,  
 imageUrl = "https://example.com/veggie-supreme-burger.jpg",  
 restaurantId = "1",  
 category = "Burger"  
 ),  
 FoodItem(  
 id = "f7",  
 name = "Chicken Maharaja Mac",  
 description = "Double chicken patty with Indian spices",  
 price = 500.0,  
 imageUrl = "https://example.com/chicken-maharaja-mac.jpg",  
 restaurantId = "1",  
 category = "Burger"  
 ),  
 FoodItem(  
 id = "f8",  
 name = "Deluxe Veggie Pizza",  
 description = "Loaded with exotic veggies and cheese",  
 price = 400.0,  
 imageUrl = "https://example.com/deluxe-veggie-pizza.jpg",  
 restaurantId = "2",  
 category = "Pizza"  
 ),  
 FoodItem(  
 id = "f9",  
 name = "Peri Peri Fries",  
 description = "Crispy fries tossed in peri peri masala",  
 price = 320.0,  
 imageUrl = "https://example.com/peri-peri-fries.jpg",  
 restaurantId = "1",  
 category = "Sides"  
 ),  
 FoodItem(  
 id = "f10",  
 name = "Cheese Burst Pizza",  
 description = "Pizza with extra cheese burst base",  
 price = 500.0,  
 imageUrl = "https://example.com/cheese-burst-pizza.jpg",  
 restaurantId = "2",  
 category = "Pizza"  
 )  
 )  
  
 NavHost(  
 navController = navController,  
 startDestination = Screen.RestaurantList.route  
 ) **{** composable(Screen.RestaurantList.route) **{** RestaurantListScreen(  
 onRestaurantClick = **{** restaurant **->** navController.navigate(Screen.RestaurantDetail.createRoute(restaurant.id))  
 **}**,  
 onCartClick = **{** navController.navigate(Screen.Cart.route) **}**,  
 cartItems = cartItems  
 )  
 **}** composable(  
 route = Screen.RestaurantDetail.route,  
 arguments = *listOf*(navArgument("restaurantId") **{** type = NavType.StringType **}**)  
 ) **{** backStackEntry **->** val restaurantId = backStackEntry.arguments?.getString("restaurantId") ?: ""  
 val restaurant = restaurants.*find* **{ it**.id == restaurantId **}** ?: return@composable  
 val items = foodItems.*filter* **{ it**.restaurantId == restaurantId **}** RestaurantDetailScreen(  
 restaurant = restaurant,  
 foodItems = items,  
 onAddToCart = **{** cartViewModel.addToCart(**it**) **}**,  
 onBack = **{** navController.popBackStack() **}** )  
 **}** composable(Screen.Cart.route) **{** CartScreen(  
 cartItems = cartItems,  
 onRemoveItem = **{** cartViewModel.removeFromCart(**it**) **}**,  
 onUpdateQuantity = **{** foodItem, quantity **->** cartViewModel.updateQuantity(foodItem, quantity)  
 **}** )  
 **}  
 }**}

# CartViewModel.kt

package com.example.foodorderapp.viewmodel  
  
import androidx.lifecycle.ViewModel  
import com.example.foodorderapp.model.CartItem  
import com.example.foodorderapp.model.FoodItem  
import kotlinx.coroutines.flow.MutableStateFlow  
import kotlinx.coroutines.flow.StateFlow  
import kotlinx.coroutines.flow.asStateFlow  
  
class CartViewModel : ViewModel() {  
 private val \_cartItems = *MutableStateFlow*<List<CartItem>>(*emptyList*())  
 val cartItems: StateFlow<List<CartItem>> = \_cartItems.*asStateFlow*()  
  
 fun addToCart(foodItem: FoodItem) {  
 val currentItems = \_cartItems.value.*toMutableList*()  
 val existingItem = currentItems.*find* **{ it**.foodItem.id == foodItem.id **}** if (existingItem != null) {  
 val index = currentItems.indexOf(existingItem)  
 currentItems[index] = existingItem.copy(quantity = existingItem.quantity + 1)  
 } else {  
 currentItems.add(CartItem(foodItem, 1))  
 }  
   
 \_cartItems.value = currentItems  
 }  
  
 fun removeFromCart(foodItem: FoodItem) {  
 \_cartItems.value = \_cartItems.value.*filter* **{ it**.foodItem.id != foodItem.id **}** }  
  
 fun updateQuantity(foodItem: FoodItem, quantity: Int) {  
 if (quantity <= 0) {  
 removeFromCart(foodItem)  
 return  
 }  
  
 val currentItems = \_cartItems.value.*toMutableList*()  
 val existingItem = currentItems.*find* **{ it**.foodItem.id == foodItem.id **}** if (existingItem != null) {  
 val index = currentItems.indexOf(existingItem)  
 currentItems[index] = existingItem.copy(quantity = quantity)  
 \_cartItems.value = currentItems  
 }  
 }  
  
 fun getTotal(): Double {  
 return \_cartItems.value.*sumOf* **{ it**.totalPrice **}** }  
}

# RestaurantCard.kt

package com.example.foodorderapp.ui.components  
  
import androidx.compose.foundation.Image  
import androidx.compose.foundation.layout.\*  
import androidx.compose.material3.\*  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.layout.ContentScale  
import androidx.compose.ui.text.font.FontWeight  
import androidx.compose.ui.unit.dp  
import coil.compose.rememberAsyncImagePainter  
import com.example.foodorderapp.model.Restaurant  
  
@Composable  
fun RestaurantCard(  
 restaurant: Restaurant,  
 onClick: () -> Unit,  
 modifier: Modifier = Modifier  
) {  
 Card(  
 onClick = onClick,  
 modifier = modifier  
 .*fillMaxWidth*()  
 .*height*(200.*dp*)  
 ) **{** Column **{** Image(  
 painter = rememberAsyncImagePainter(restaurant.imageUrl),  
 contentDescription = restaurant.name,  
 modifier = Modifier  
 .*fillMaxWidth*()  
 .*height*(120.*dp*),  
 contentScale = ContentScale.Crop  
 )  
   
 Column(  
 modifier = Modifier  
 .*padding*(16.*dp*)  
 .*fillMaxWidth*()  
 ) **{** Text(  
 text = restaurant.name,  
 style = MaterialTheme.typography.titleLarge,  
 fontWeight = FontWeight.Bold  
 )  
   
 Spacer(modifier = Modifier.*height*(4.*dp*))  
   
 Row(  
 modifier = Modifier.*fillMaxWidth*(),  
 horizontalArrangement = Arrangement.SpaceBetween,  
 verticalAlignment = Alignment.CenterVertically  
 ) **{** Text(  
 text = "⭐ ${restaurant.rating}",  
 style = MaterialTheme.typography.bodyMedium  
 )  
 Text(  
 text = restaurant.deliveryTime,  
 style = MaterialTheme.typography.bodyMedium  
 )  
 **}  
 }  
 }  
 }**}