TITLE:- Android Based Furniture Shopping Project

The Android Based Furniture Shopping Project aims to provide users with a convenient platform to browse and purchase furniture items using their mobile devices. This project leverages the popularity and accessibility of Android devices to create a seamless shopping experience for users interested in furniture.

OBJECTIVES:-

- Develop an intuitive Android application for browsing and purchasing furniture.
- Provide users with a wide range of furniture categories and items to choose from.
- Implement secure payment gateways for safe transactions.
- Incorporate features for users to track orders and manage their shopping carts.
- Ensure a user-friendly interface and smooth navigation throughout the application.

FEATURES:-

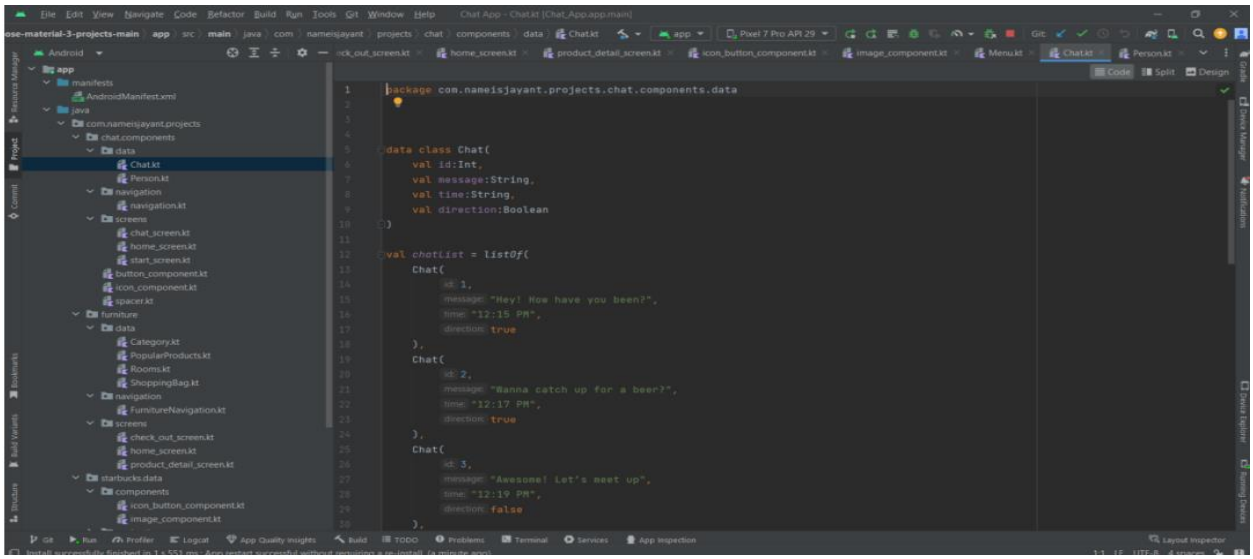
- Intuitive User Interface: Design an easy-to-use interface for seamless navigation and enhanced user experience.
- Comprehensive Product Catalog: Offer a diverse range of furniture categories and items with detailed descriptions and images.
- Search and Filter Functionality: Enable users to quickly find desired items by implementing search and filter options based on various criteria such as category, price range, and materials.
- Secure Payment Integration: Integrate trusted payment gateways to ensure safe and encrypted transactions, supporting multiple payment methods.
- Cart Management: Allow users to add, remove, and modify items in their shopping carts before checkout, enhancing flexibility and convenience.
- Order Tracking: Provide real-time updates on order status, from processing to delivery, enabling users to track their purchases seamlessly.
- User Account Management: Enable users to create accounts, login securely, and manage their profiles, including order history and preferences.
- Wishlist Feature: Allow users to add favorite items to a wishlist for future reference or purchase.
- Ratings and Reviews: Implement a system for users to leave feedback and ratings on purchased items, fostering transparency and trust within the community.
- Push Notifications: Send timely notifications to users regarding order updates, promotions, and personalized recommendations, keeping them engaged and informed.

DESCRIPTION:-

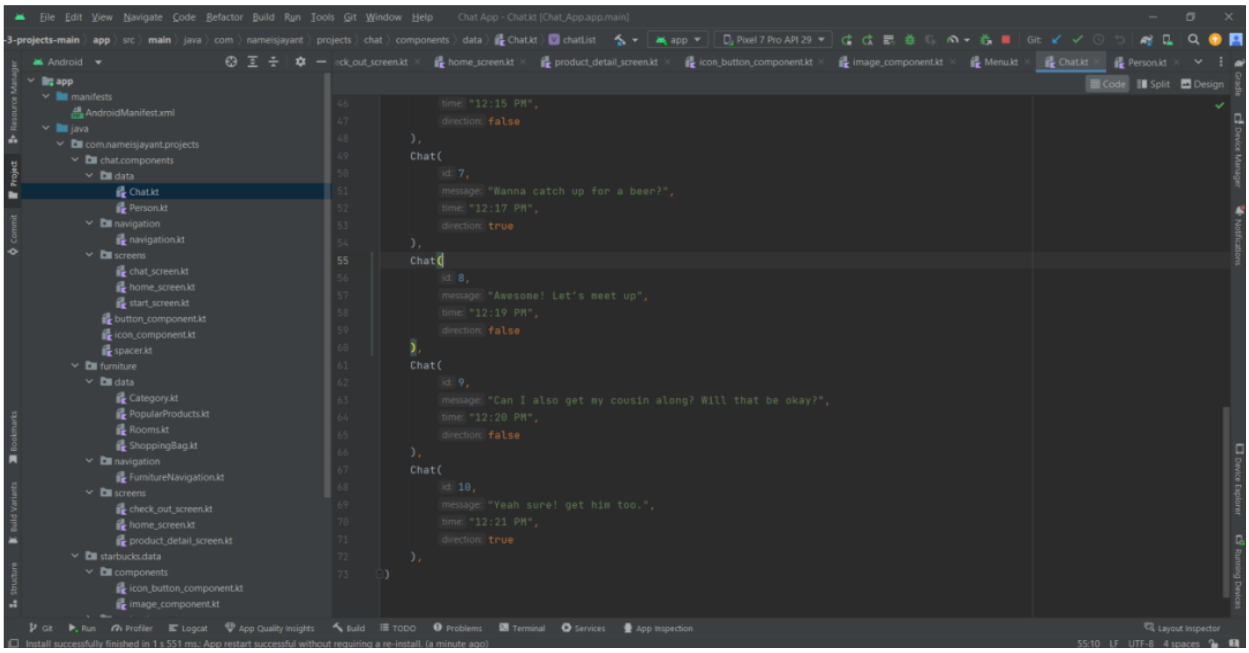
- Intuitive User Interface: The Android Furniture Shopping App boasts a visually appealing and intuitive user interface designed to enhance user engagement and satisfaction. With clear navigation menus, visually pleasing layouts, and intuitive controls, users can easily browse through the extensive catalog of furniture items without any hassle.
- Comprehensive Product Catalog: From elegant chairs to stylish sofas, sturdy tables to cozy beds, the app offers a comprehensive collection of furniture items to cater to diverse tastes and preferences. Each product is meticulously curated with detailed descriptions, high-quality images, and specifications to provide users with all the necessary information they need to make informed purchasing decisions.

- Search and Filter Functionality: Finding the perfect furniture item is made effortless with the app's powerful search and filter functionality. Users can quickly search for specific items using keywords or apply filters based on categories, price range, materials, and more, enabling them to narrow down their choices and discover exactly what they're looking for in no time.
- Secure Payment Integration: The app prioritizes the security of user transactions by integrating trusted and secure payment gateways. Whether users prefer to pay via credit/debit cards, net banking, or digital wallets, they can rest assured that their payment information is encrypted and protected throughout the checkout process.
- Cart Management: With the cart management feature, users can easily add their favorite furniture items to their shopping carts, remove any unwanted items, and adjust quantities as needed before proceeding to checkout. This flexibility allows users to review and finalize their purchases with confidence.
- Order Tracking: Keeping users informed every step of the way, the app provides real-time updates on the status of their orders, from processing to delivery. Users can track their purchases seamlessly, knowing exactly when to expect their furniture items to arrive at their doorstep.
- User Account Management: By creating accounts and logging in securely, users can enjoy personalized experiences and access features such as order history, saved preferences, and wishlist management. This enhances user engagement and fosters a sense of belonging within the app community.
- Wishlist Feature: Users can save their favorite furniture items to a wishlist for future reference or purchase. Whether they're planning their next home makeover or simply browsing for inspiration, the wishlist feature allows users to keep track of items they love and revisit them at any time.
- Ratings and Reviews: To promote transparency and trust, the app enables users to leave feedback and ratings on purchased items. This not only helps other users make informed decisions but also provides valuable insights for sellers to improve their products and services.
- Push Notifications: With timely push notifications, users stay informed about order updates, promotions, and personalized recommendations, ensuring they never miss out on exciting deals or new arrivals. This proactive approach to communication keeps users engaged and encourages repeat visits to the app.

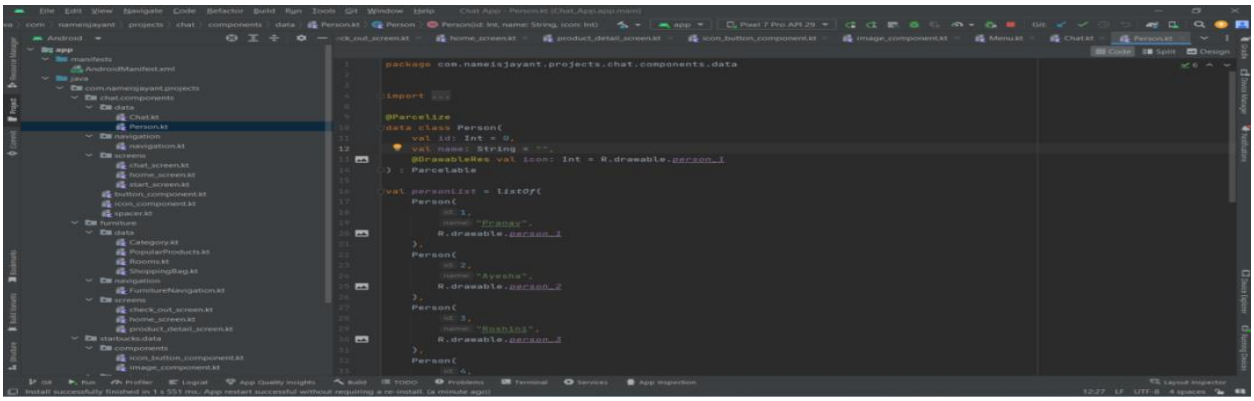
SCREENSHOTS OF CODE USED



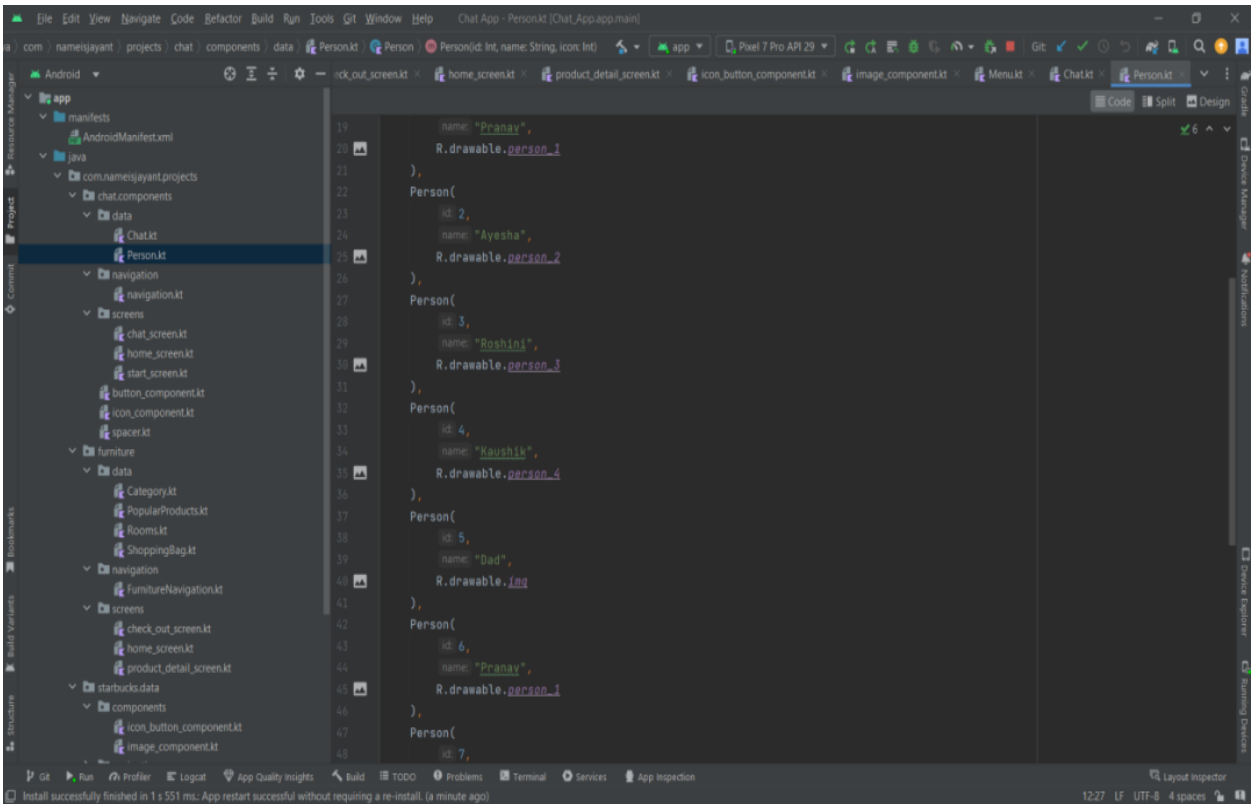
```
package com.nameisjayant.projects.chat.components.data


data class Chat(
    val id:Int,
    val message:String,
    val time:String,
    val direction:Boolean
)

val chatList = listOf(
    Chat(
        id: 1,
        message: "Hey! How have you been?",
        time: "12:15 PM",
        direction: true
    ),
    Chat(
        id: 2,
        message: "Wanna catch up for a beer?",
        time: "12:17 PM",
        direction: true
    ),
    Chat(
        id: 3,
        message: "Awesome! Let's meet up",
        time: "12:19 PM",
        direction: false
    ),
```

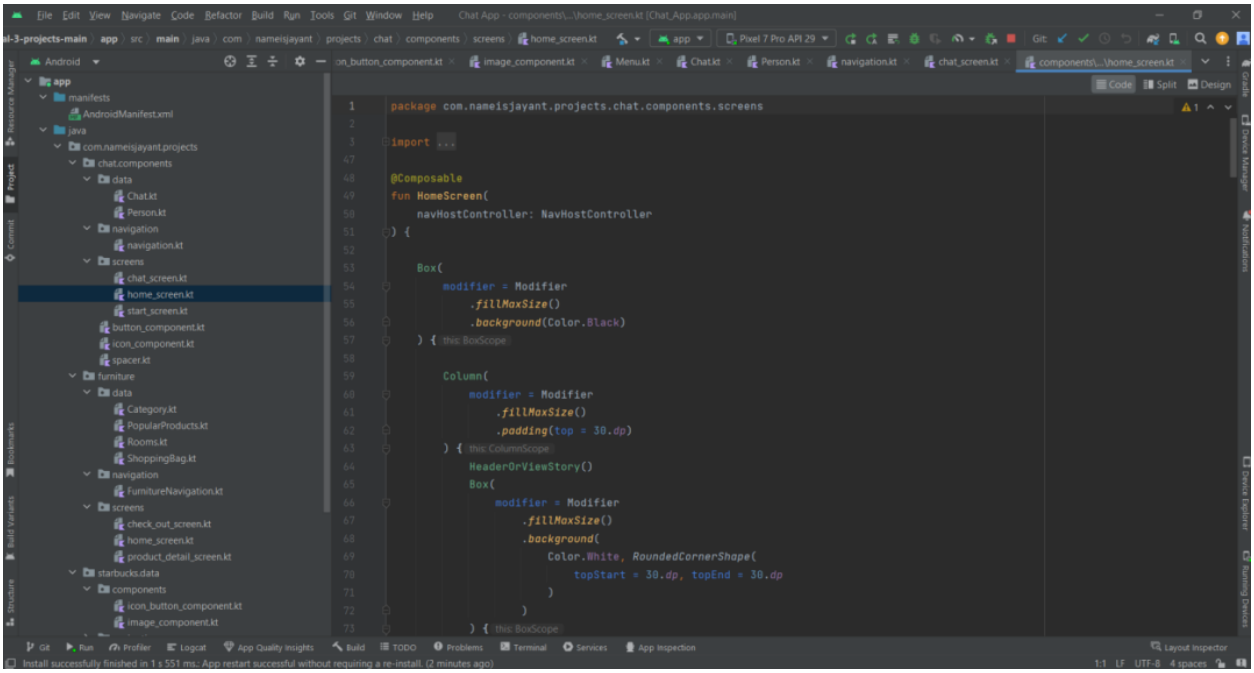


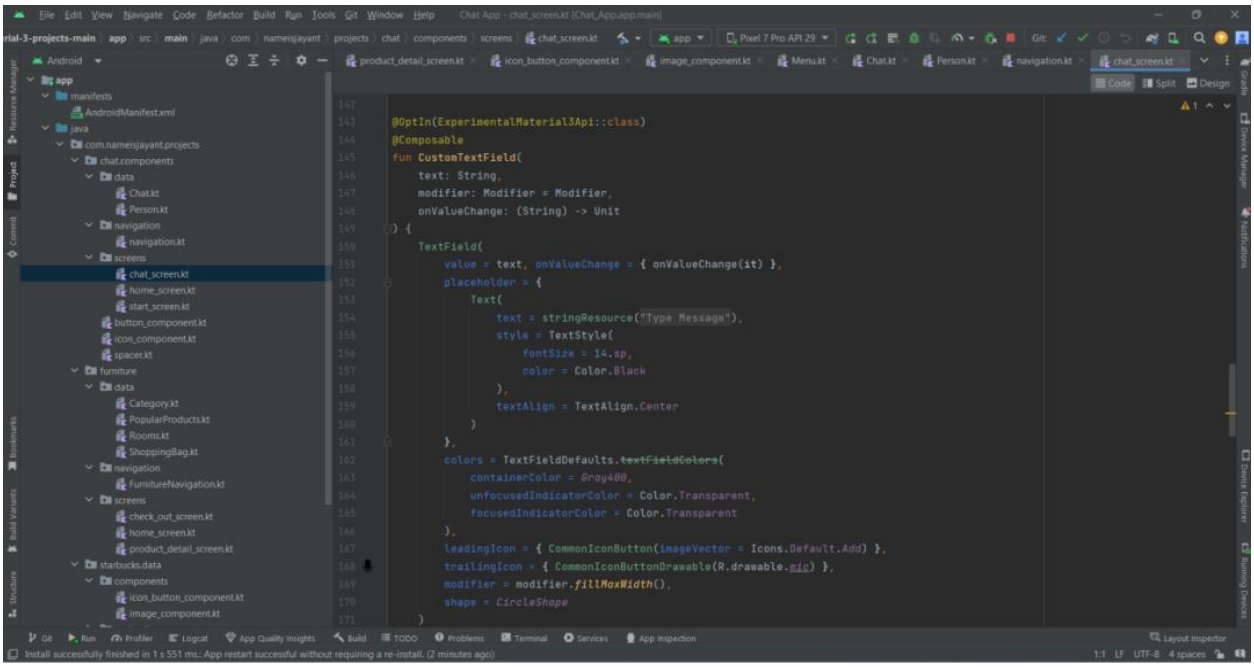
```
        time: "12:15 PM",
        direction: false
    ),
    Chat(
        id: 7,
        message: "Wanna catch up for a beer?",
        time: "12:17 PM",
        direction: true
    ),
    Chat(
        id: 8,
        message: "Awesome! Let's meet up",
        time: "12:19 PM",
        direction: false
    ),
    Chat(
        id: 9,
        message: "Can I also get my cousin along? Will that be okay?",
        time: "12:20 PM",
        direction: false
    ),
    Chat(
        id: 10,
        message: "Yeah sure! get him too.",
        time: "12:21 PM",
        direction: true
    ),
)
```



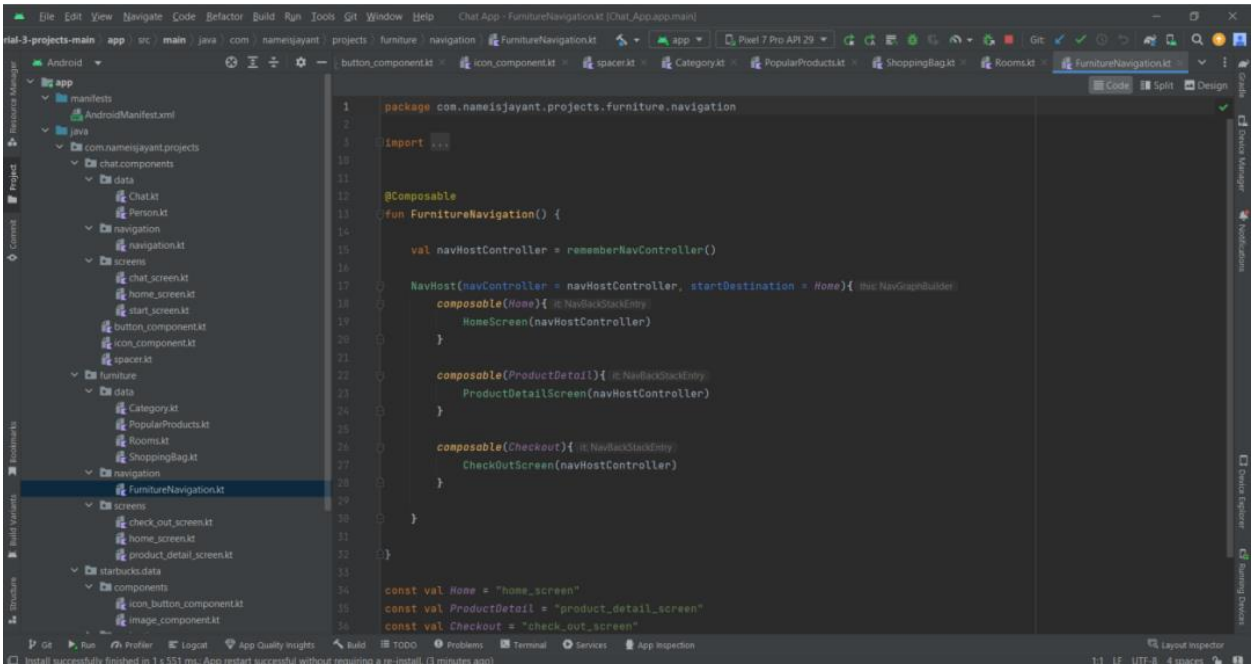
```
package com.nameisjayant.projects.chat.components.data


import ...

@Parcelize
data class Person(
    val id: Int = 0,
    val name: String = "",
    @DrawableRes val icon: Int = R.drawable.person_1
) : Parcelable

val personList = listOf(
    Person(
        id: 1,
        name: "Pranav",
        R.drawable.person_1
    ),
    Person(
        id: 2,
        name: "Ayesha",
        R.drawable.person_2
    ),
    Person(
        id: 3,
        name: "Roshini",
        R.drawable.person_3
    ),
    Person(
```


Person.kt
Person(
id: 2,
name: "Ayesha",
R.drawable.person_2
Person(
id: 3,
name: "Roshini",
R.drawable.person_3
Person(
id: 4,
name: "Kaushik",
R.drawable.person_4
Person(
id: 5,
name: "Dad",
R.drawable.ima
Person(
id: 6,
name: "Pranav",
R.drawable.person_1
Person(
id: 7,

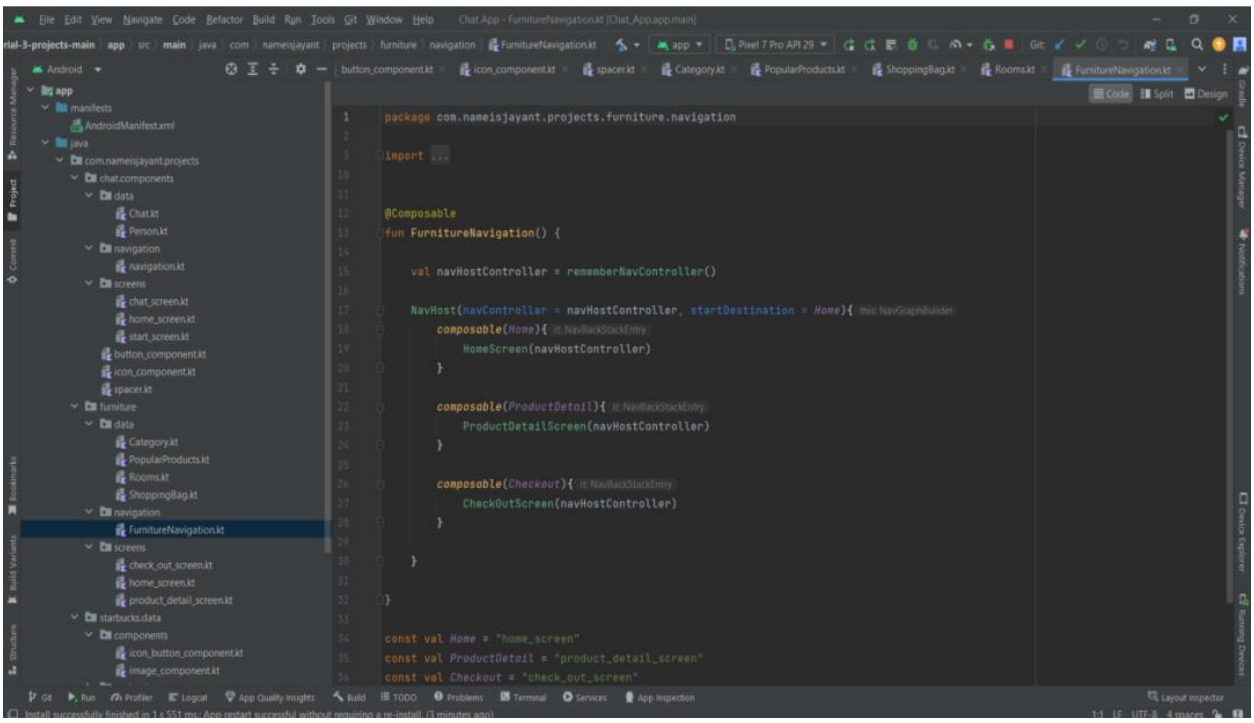

home_screen.kt
package com.nameisjayant.projects.chat.components.screens
import ...
@Composable
fun HomeScreen(
navHostController: NavHostController
) {
Box(
modifier = Modifier
.fillMaxSize()
.background(Color.Black)
) { this: BoxScope
Column(
modifier = Modifier
.fillMaxSize()
.padding(top = 30.dp)
) { this: ColumnScope
HeaderOrViewStory()
Box(
modifier = Modifier
.fillMaxSize()
.background(
Color.White, RoundedCornerShape(
topStart = 30.dp, topEnd = 30.dp
)
)
) { this: BoxScope

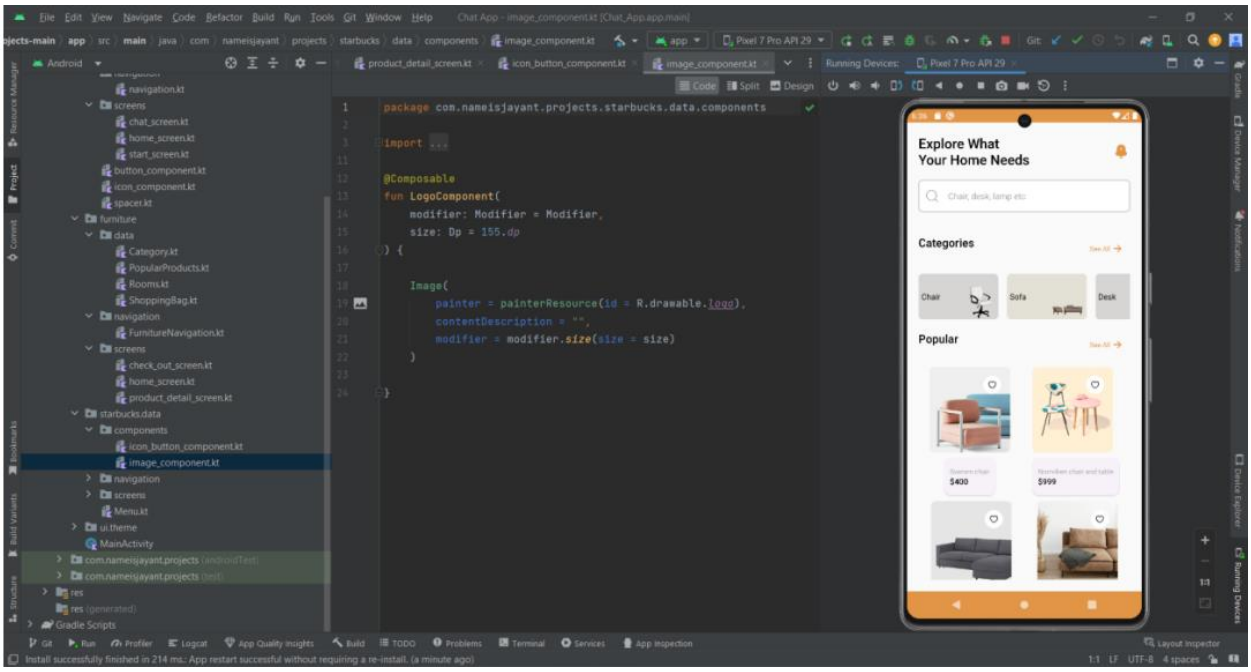


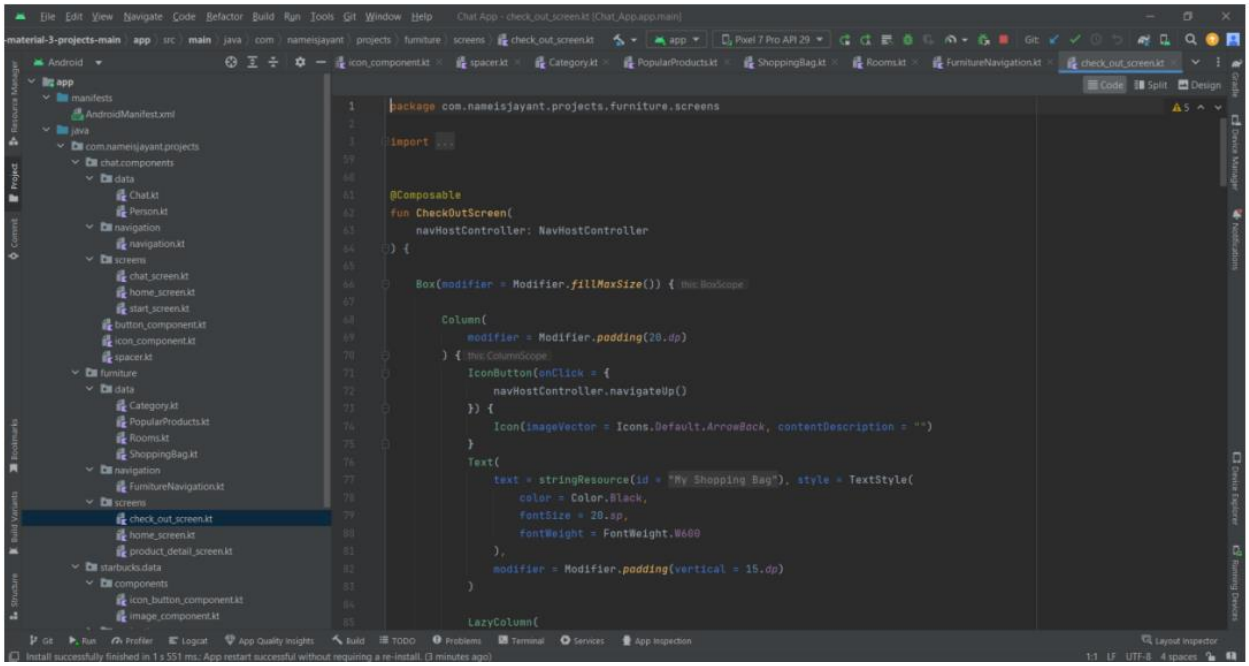
```kotlin
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun CustomTextField(
    text: String,
    modifier: Modifier = Modifier,
    onValueChange: (String) -> Unit
) {
    TextField(
        value = text, onValueChange = { onValueChange(it) },
        placeholder = {
            Text(
                text = stringResource("Type Message"),
                style = TextStyle(
                    fontSize = 14.sp,
                    color = Color.Black
                ),
                textAlign = TextAlign.Center
            )
        },
        colors = TextFieldDefaults.textFieldColors(
            containerColor = Gray400,
            unfocusedIndicatorColor = Color.Transparent,
            focusedIndicatorColor = Color.Transparent
        ),
        leadingIcon = { CommonIconButton(imageVector = Icons.Default.Add) },
        trailingIcon = { CommonIconButtonDrawable(R.drawable.mic) },
        modifier = modifier.fillMaxWidth(),
        shape = CircleShape
    )
```

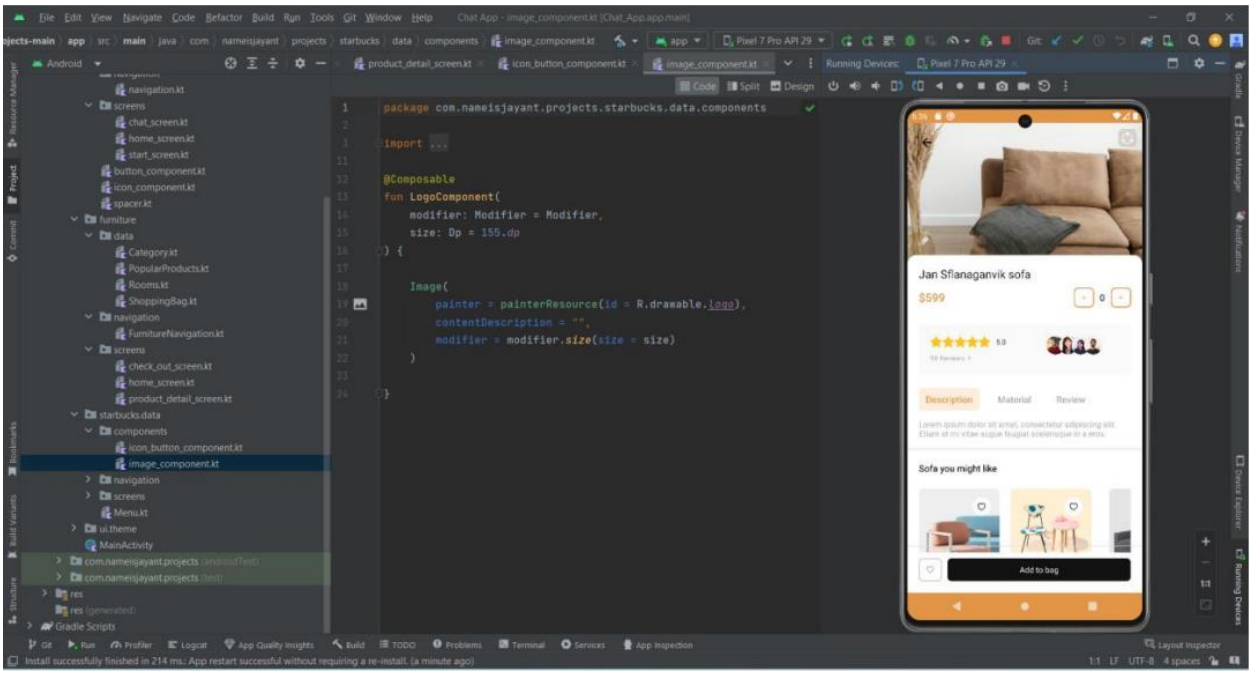


```kotlin
package com.nameisjayant.projects.furniture.navigation

import ...


@Composable
fun FurnitureNavigation() {

    val navHostController = rememberNavController()

    NavHost(navController = navHostController, startDestination = Home){
        composable(Home){
            HomeScreen(navHostController)
        }

        composable(ProductDetail){
            ProductDetailScreen(navHostController)
        }

        composable(Checkout){
            CheckOutScreen(navHostController)
        }

    }

}

const val Home = "home_screen"
const val ProductDetail = "product_detail_screen"
const val Checkout = "check_out_screen"
```

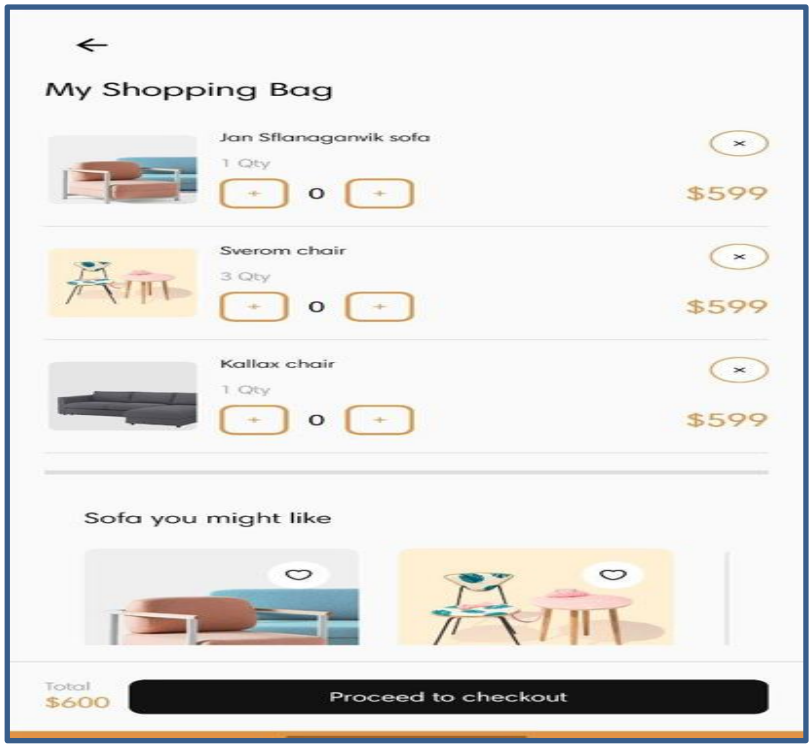

package com.nameisjayant.projects.furniture.navigation
import ...
@Composable
fun FurnitureNavigation() {
val navHostController = rememberNavController()
NavHost(navController = navHostController, startDestination = Home){
composable(Home){
HomeScreen(navHostController)
}
composable(ProductDetail){
ProductDetailScreen(navHostController)
}
composable(Checkout){
CheckOutScreen(navHostController)
}
}
}
const val Home = "home_screen"
const val ProductDetail = "product_detail_screen"
const val Checkout = "check_out_screen"

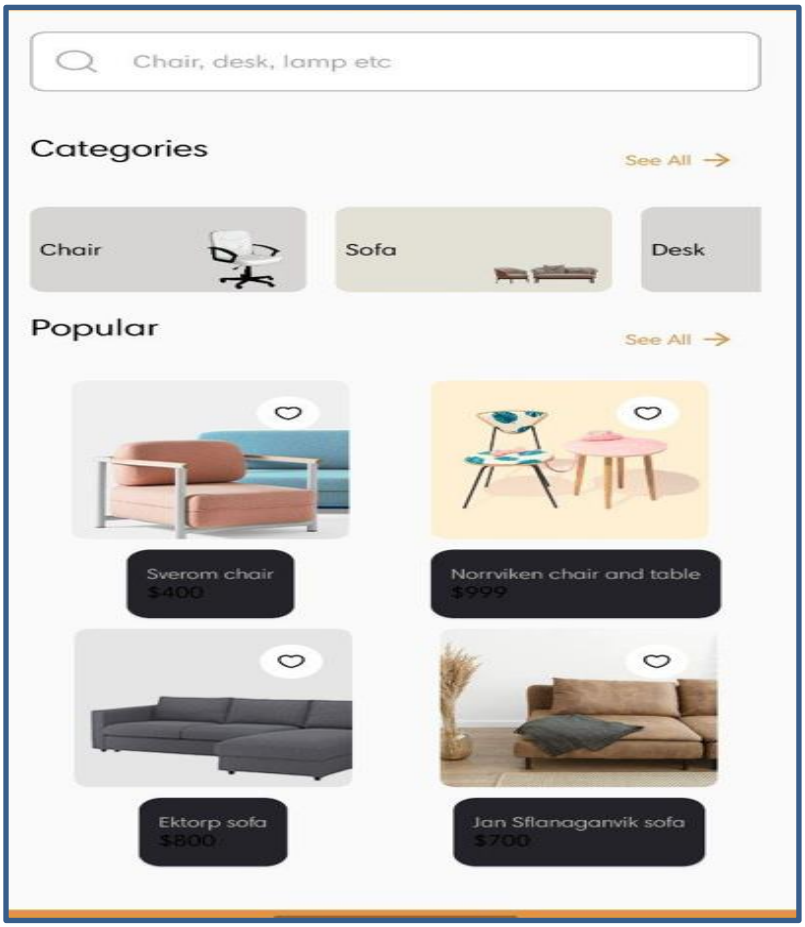

package com.nameisjayant.projects.starbucks.data.components
import ...
@Composable
fun LogoComponent(
modifier: Modifier = Modifier,
size: Dp = 155.dp
) {
Image(
painter = painterResource(id = R.drawable.logo),
contentDescription = "",
modifier = modifier.size(size = size)
)
}
Explore What
Your Home Needs
Categories
Chair
Sofa
Desk
Popular
$400
$999


package com.nameisjayant.projects.furniture.screens
import ...
@Composable
fun CheckOutScreen(
navHostController: NavHostController
) {
Box(modifier = Modifier.fillMaxSize()) {
Column(
modifier = Modifier.padding(20.dp)
) {
IconButton(onClick = {
navHostController.navigateUp()
}) {
Icon(imageVector = Icons.Default.ArrowBack, contentDescription = "")
}
Text(
text = stringResource(id = "My Shopping Bag"), style = TextStyle(
color = Color.Black,
fontSize = 20.sp,
fontWeight = FontWeight.W600
),
modifier = Modifier.padding(vertical = 15.dp)
)
LazyColumn(


package com.nameisjayant.projects.starbucks.data.components
import ...
@Composable
fun LogoComponent(
modifier: Modifier = Modifier,
size: Dp = 155.dp
) {
Image(
painter = painterResource(id = R.drawable.logo),
contentDescription = "",
modifier = modifier.size(size = size)
)
}
Jan Sflanaganvik sofa
$599
Description
Material
Review
Sofa you might like
Add to bag

SCREENSHOTS OF OUTPUT





My Shopping Bag

Jan Sflanaganvik sofa

1 Qty

+ 0 +

$599

Sverom chair

3 Qty

+ 0 +

$599

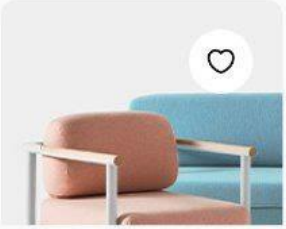
Kallax chair

1 Qty

+ 0 +

$599

Sofa you might like



Total
$600

Proceed to checkout