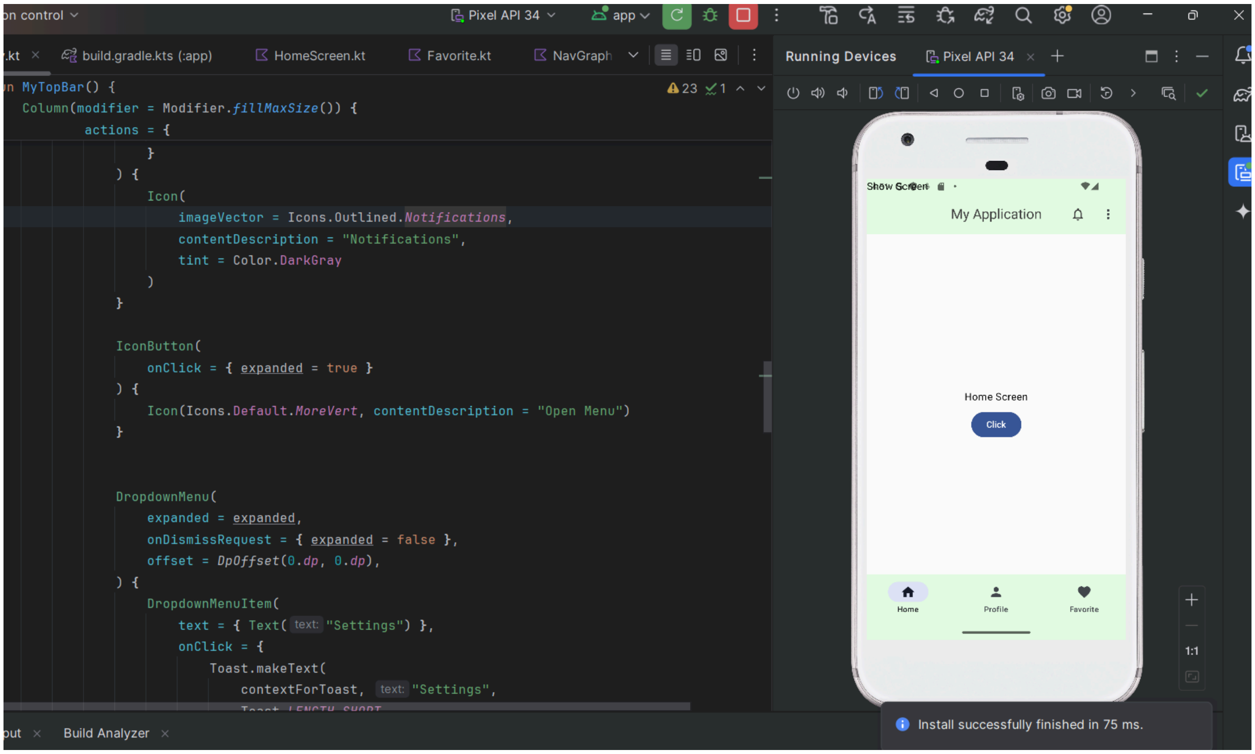
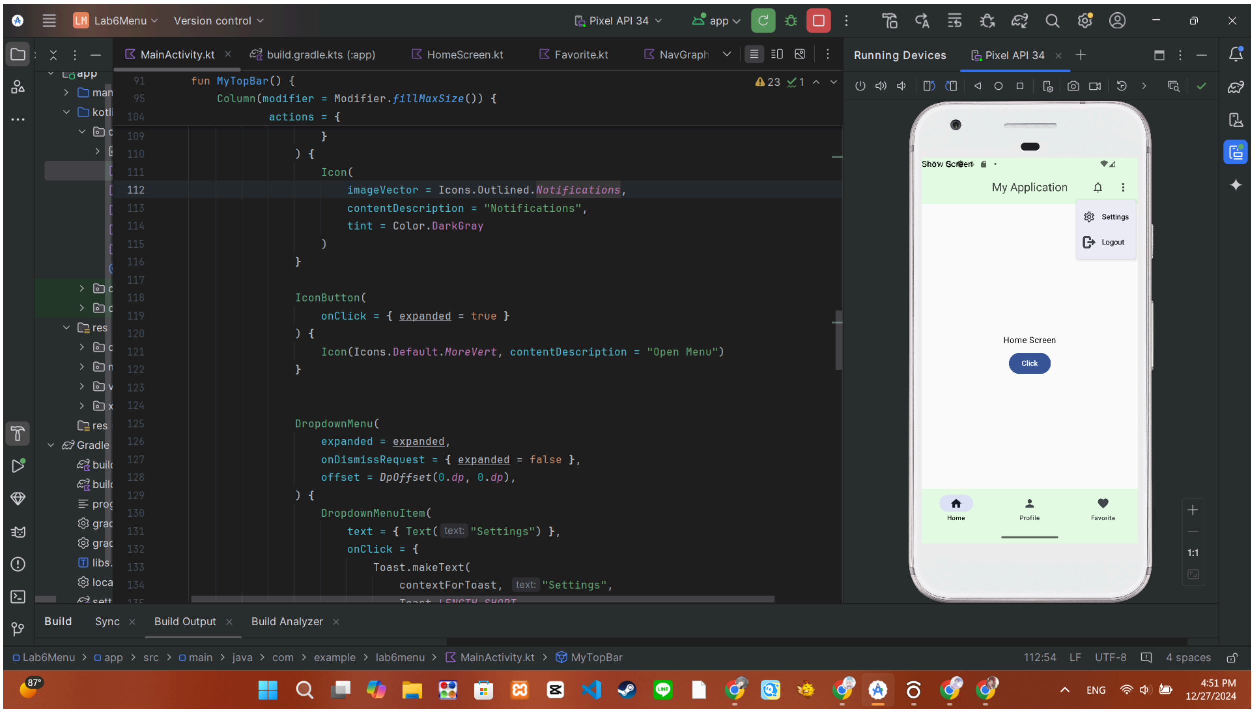
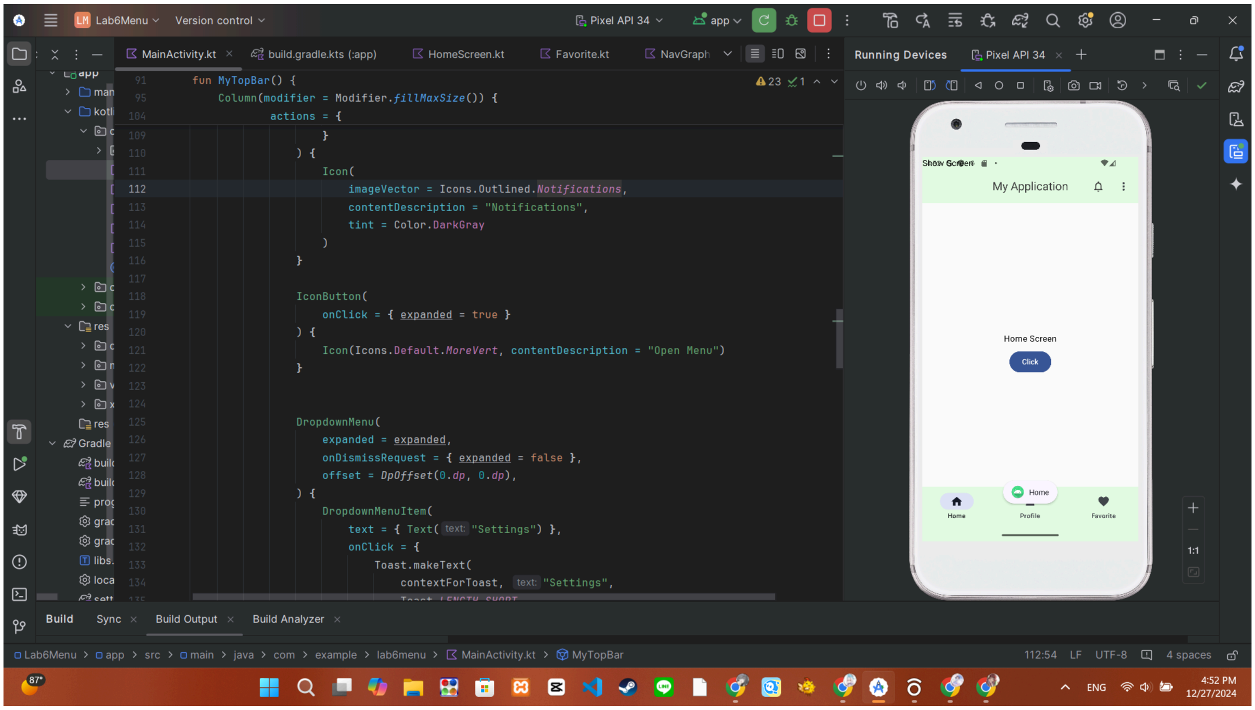


```
n MyTopBar() {
    Column(modifier = Modifier.fillMaxSize()) {
            actions = {
                    }
                ) {
                    Icon(
                        imageVector = Icons.Outlined.Notifications,
                        contentDescription = "Notifications",
                        tint = Color.DarkGray
                    )
                }

                IconButton(
                    onClick = { expanded = true }
                ) {
                    Icon(Icons.Default.MoreVert, contentDescription = "Open Menu")
                }


                DropdownMenu(
                    expanded = expanded,
                    onDismissRequest = { expanded = false },
                    offset = DpOffset(0.dp, 0.dp),
                ) {
                    DropdownMenuItem(
                        text = { Text( text: "Settings") },
                        onClick = {
                            Toast.makeText(
                                contextForToast,  text: "Settings",
                                Toast.LENGTH_SHORT
```

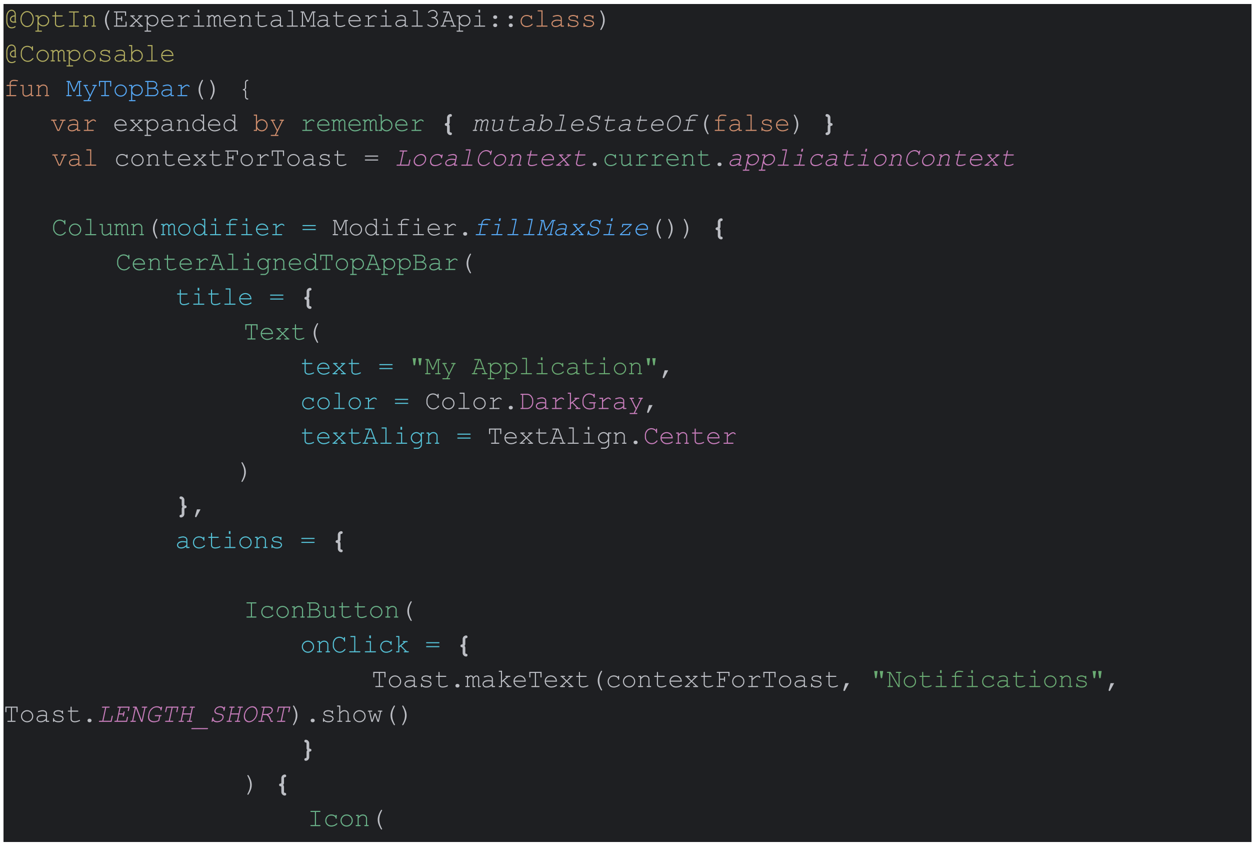
Install successfully finished in 75 ms.



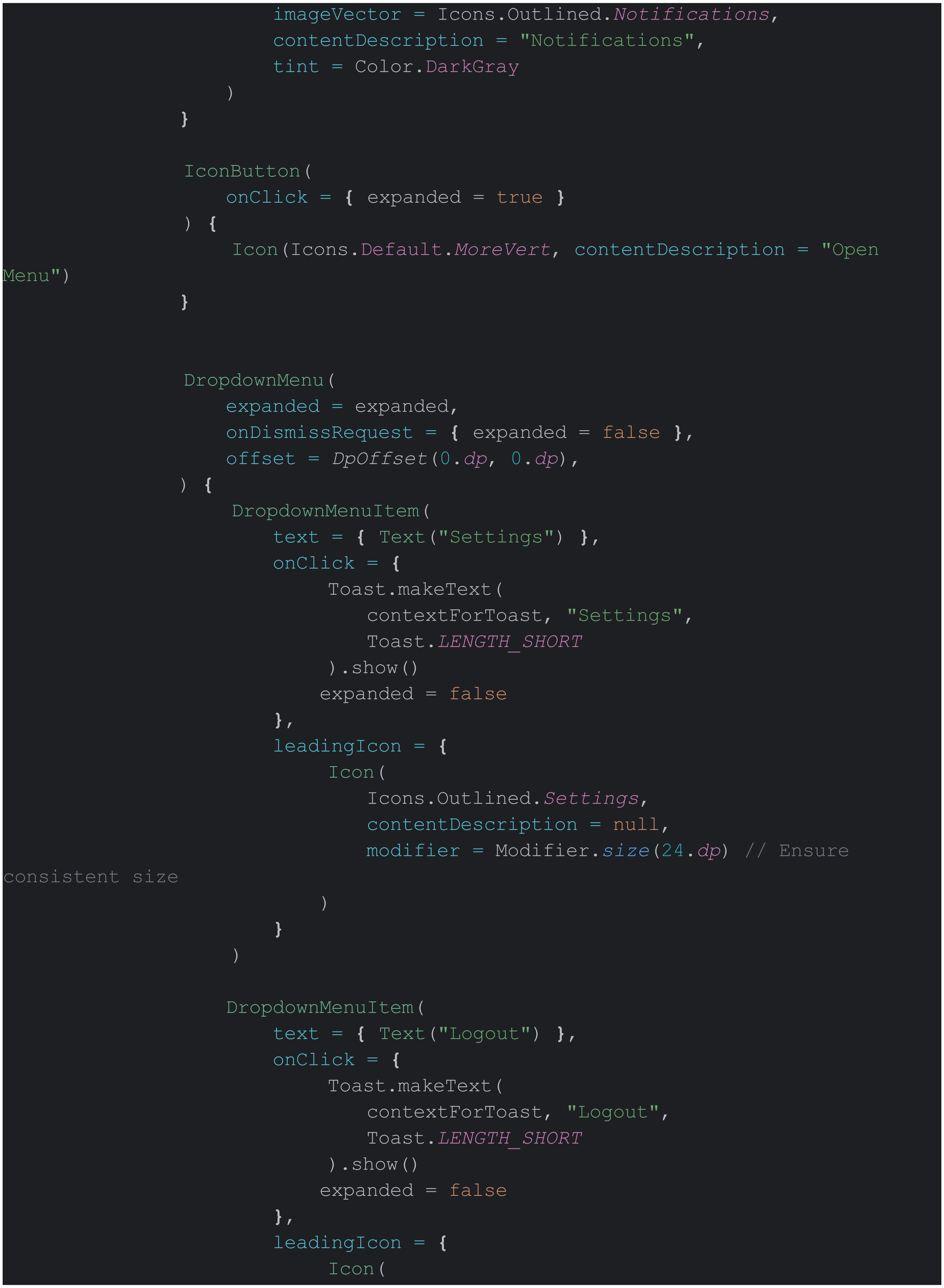
```
    fun MyTopBar() {
        Column(modifier = Modifier.fillMaxSize()) {
                actions = {
                        }
                    ) {
                        Icon(
                            imageVector = Icons.Outlined.Notifications,
                            contentDescription = "Notifications",
                            tint = Color.DarkGray
                        )
                    }

                    IconButton(
                        onClick = { expanded = true }
                    ) {
                        Icon(Icons.Default.MoreVert, contentDescription = "Open Menu")
                    }


                    DropdownMenu(
                        expanded = expanded,
                        onDismissRequest = { expanded = false },
                        offset = DpOffset(0.dp, 0.dp),
                    ) {
                        DropdownMenuItem(
                            text = { Text( text: "Settings") },
                            onClick = {
                                Toast.makeText(
                                    contextForToast,  text: "Settings",
                                    Toast.LENGTH_SHORT
```



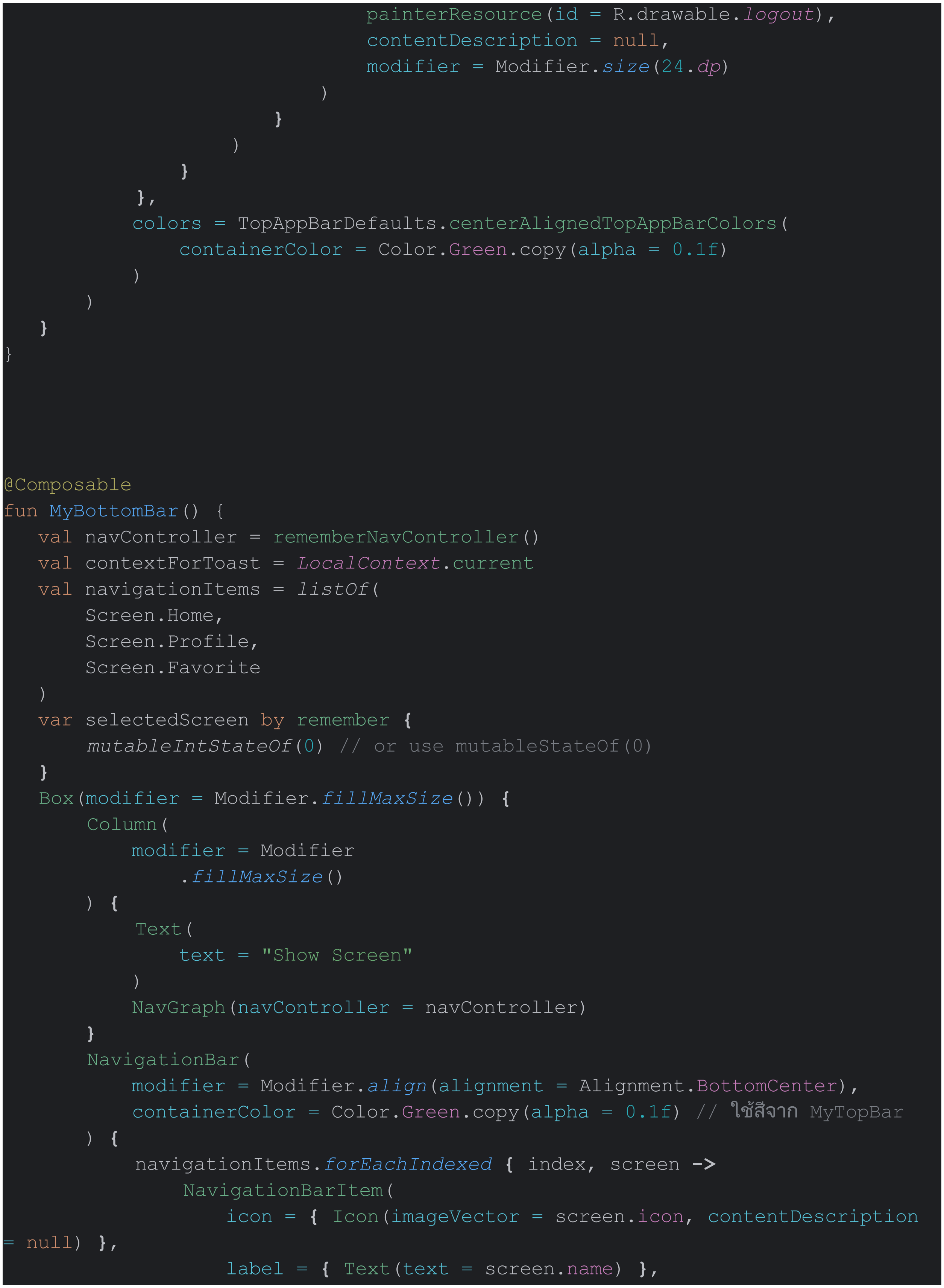
```
fun MyTopBar() {
    Column(modifier = Modifier.fillMaxSize()) {
            actions = {
                        }
                    ) {
                        Icon(
                            imageVector = Icons.Outlined.Notifications,
                            contentDescription = "Notifications",
                            tint = Color.DarkGray
                        )
                    }

                    IconButton(
                        onClick = { expanded = true }
                    ) {
                        Icon(Icons.Default.MoreVert, contentDescription = "Open Menu")
                    }


                    DropdownMenu(
                        expanded = expanded,
                        onDismissRequest = { expanded = false },
                        offset = DpOffset(0.dp, 0.dp),
                    ) {
                        DropdownMenuItem(
                            text = { Text( text: "Settings") },
                            onClick = {
                                Toast.makeText(
                                    contextForToast,  text: "Settings",
```



```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun MyTopBar() {
   var expanded by remember { mutableStateOf(false) }
   val contextForToast = LocalContext.current.applicationContext

   Column(modifier = Modifier.fillMaxSize()) {
       CenterAlignedTopAppBar(
           title = {
               Text(
                   text = "My Application",
                   color = Color.DarkGray,
                   textAlign = TextAlign.Center
               )
           },
           actions = {

               IconButton(
                   onClick = {
                       Toast.makeText(contextForToast, "Notifications",
Toast.LENGTH_SHORT).show()
                   }
               ) {
                   Icon(
```



```
                        imageVector = Icons.Outlined.Notifications,
                        contentDescription = "Notifications",
                        tint = Color.DarkGray
                    )
                }

                IconButton(
                    onClick = { expanded = true }
                ) {
                    Icon(Icons.Default.MoreVert, contentDescription = "Open
Menu")
                }


                DropdownMenu(
                    expanded = expanded,
                    onDismissRequest = { expanded = false },
                    offset = DpOffset(0.dp, 0.dp),
                ) {
                    DropdownMenuItem(
                        text = { Text("Settings") },
                        onClick = {
                            Toast.makeText(
                                contextForToast, "Settings",
                                Toast.LENGTH_SHORT
                            ).show()
                            expanded = false
                        },
                        leadingIcon = {
                            Icon(
                                Icons.Outlined.Settings,
                                contentDescription = null,
                                modifier = Modifier.size(24.dp) // Ensure
consistent size
                            )
                        }
                    )

                    DropdownMenuItem(
                        text = { Text("Logout") },
                        onClick = {
                            Toast.makeText(
                                contextForToast, "Logout",
                                Toast.LENGTH_SHORT
                            ).show()
                            expanded = false
                        },
                        leadingIcon = {
                            Icon(
```



```
                                painterResource(id = R.drawable.logout),
                                contentDescription = null,
                                modifier = Modifier.size(24.dp)
                            )
                        }
                    )
                }
            },
            colors = TopAppBarDefaults.centerAlignedTopAppBarColors(
                containerColor = Color.Green.copy(alpha = 0.1f)
            )
        )
    }
}



@Composable
fun MyBottomBar() {
    val navController = rememberNavController()
    val contextForToast = LocalContext.current
    val navigationItems = listOf(
        Screen.Home,
        Screen.Profile,
        Screen.Favorite
    )
    var selectedScreen by remember {
        mutableIntStateOf(0) // or use mutableStateOf(0)
    }
    Box(modifier = Modifier.fillMaxSize()) {
        Column(
            modifier = Modifier
                .fillMaxSize()
        ) {
            Text(
                text = "Show Screen"
            )
            NavGraph(navController = navController)
        }
        NavigationBar(
            modifier = Modifier.align(alignment = Alignment.BottomCenter),
            containerColor = Color.Green.copy(alpha = 0.1f) // ใช้สีจาก MyTopBar
        ) {
            navigationItems.forEachIndexed { index, screen ->
                NavigationBarItem(
                    icon = { Icon(imageVector = screen.icon, contentDescription
= null) },
                    label = { Text(text = screen.name) },
                    selected = (selectedScreen == index),
                    onClick = {
                         selectedScreen = index
                        navController.navigate(screen.route)
                        Toast.makeText(contextForToast, screen.name,
Toast.LENGTH_SHORT).show()
                    }
                )
            }
        }
    }
}
```