

Snacks Ordering App

DESCRIPTION

Snacks Ordering App

- **Personalized Recommendations:** Suggests snacks based on user preferences and history.
- **Real-Time Order Tracking:** Tracks orders from placement to delivery.
- **Secure Payments:** Uses **Stripe** or **Razorpay** for safe transactions.
- **Dynamic Reports:** Visualizes spending trends via **Chart.js**.
- **Cloud Syncing:** Saves preferences and order history with **Firebase**.
- **Push Notifications:** Alerts for deals, new snacks, and updates.

Development Tools

- **Figma** for design.
- **React Native/Flutter** for frontend.
- **Firebase/PostgreSQL** for backend.

Competitive Edge

- Snack-focused, with curated menus and faster delivery than broad platforms like **Zomato**.

Main Activity.java:

```
package com.example.snackordering
import android.annotation.SuppressLint
import android.content.Contextimport android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.annotation.DrawableRes
import androidx.annotation.StringResimport androidx.compose.foundation.Image
import androidx.compose.foundation.background
```

```
import androidx.compose.foundation.layout.  
import androidx.compose.foundation.shape.CircleShape  
import androidx.compose.foundation.shape.RoundedCornerShape  
import androidx.compose.material.  
import androidx.compose.material.icons.Icons  
import androidx.compose.material.icons.filled.  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Alignmentimport androidx.compose.ui.Modifier
```

```
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.expensetracker.ui.theme.Snacks Ordering
Theme
```

```
class MainPage : ComponentActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView {  
            SnackOrderingTheme {  
                // A surface container using the 'background' color from the theme  
                Surface(  
                    modifier = Modifier.fillMaxSize(),  
                    color = MaterialTheme.colors.background  
                ) {  
                    FinalView(this)  
                    val context = LocalContext.current  
                    //PopularFoodColumn(context)  
                }  
            }  
        }  
    }  
}
```

```
@Composable
fun TopPart() {
```

```
    Row(
        modifier = Modifier
            .fillMaxWidth()
            .background(Color(0xFFECF0F0)), Arrangement.SpaceBetween
    ) {
        Icon(
            imageVector = Icons.Default.Add, contentDescription = "Menu Icon",
            Modifier

                .clip(CircleShape)
                .size(40.dp),
            tint = Color.Black,
        )
        Column(horizontalAlignment = Alignment.CenterHorizontally) {
            Text(text = "Location", style = MaterialTheme.typography.subtitle1, color = Color.Black)
            Row {
                Icon(
                    imageVector = Icons.Default.LocationOn,
                    contentDescription = "Location",
                    tint = Color.Red,
                )
            }
        }
    }
}
```

```
Text(text = "coimbatore" , color = Color.Black)
    }

    }
    Icon(
        imageVector = Icons.Default.Notifications, contentDescription = "Notification Icon",

        Modifier
            .size(45.dp),
            tint = Color.Black,
    )
}
}
```

```
@Composable
fun CardPart() {
    Card(modifier = Modifier.size(width = 310.dp, height = 150.dp), RoundedCornerShape(20.dp)) {
        Row(modifier = Modifier.padding(10.dp), Arrangement.SpaceBetween) {
            Column(verticalArrangement = Arrangement.spacedBy(12.dp)) {
                Text(text = "TODAY ONLY!!!!")
                Text(text = "Eat and Enjoy")
                Text(text = "Up to 85%", style = MaterialTheme.typography.h5)
                Button(onClick = {}, colors = ButtonDefaults.buttonColors(Color.White)) {
                    Text(text = "Claim voucher", color = MaterialTheme.colors.surface)
                }
            }
        }
    }
}
```



```
}  
    Image(  
        painter = painterResource(id = R.drawable.all_in_one),  
        contentDescription = "Food Image", Modifier.size(width = 100.dp, height = 200.dp)  
    )  
}  
}  
}
```

```
@Composable  
fun PopularFood(  
    @DrawableRes drawable: Int,  
    @StringRes text1: Int,  
    context: Context  
) {  
    Card(  
        modifier = Modifier  
            .padding(top = 20.dp, bottom = 20.dp, start = 65.dp)  
            .width(250.dp)  
  
    ) {
```

```
Column(  
    verticalArrangement = Arrangement.Top,  
    horizontalAlignment = Alignment.CenterHorizontally  
) {  
    Spacer(modifier = Modifier.padding(vertical = 5.dp))  
    Row(  
        modifier = Modifier  
            .fillMaxWidth(0.7f), Arrangement.End  
    ) {  
        Icon(  
            imageVector = Icons.Default.Star,  
            contentDescription = "Star Icon",  
            tint = Color.Yellow  
        )  
        Text(text = "4.3", fontWeight = FontWeight.Black)  
    }  
    Image(  
        painter = painterResource(id = drawable),  
        contentDescription = "Food Image",  
        contentScale = ContentScale.Crop,  
        modifier = Modifier  
            .size(100.dp)  
            .clip(CircleShape)  
    )  
}
```

)

```
Text(text = stringResource(id = text1), fontWeight = FontWeight.Bold)
Row(modifier = Modifier.fillMaxWidth(0.7f), Arrangement.SpaceBetween) {
    /TODO Implement Prices for each card/
    Text(
        text = "$50",
        style = MaterialTheme.typography.h6,
        fontWeight = FontWeight.Bold,
        fontSize = 18.sp
    )

    IconButton(onClick = {

        //var no=FoodList.lastIndex;
        //Toast.
        val intent = Intent1(context, TargetActivity::class.java)
        context.startActivity(intent)

    }) {
        Icon(
            imageVector = Icons.Default.ShoppingCart,
            contentDescription = "shopping cart",
        )
    }
}
}
```

```
private val FoodList = listOf(  
    R.drawable.sandwitch to R.string.sandwich,  
    R.drawable.all_in_one to R.string.burgers,  
    R.drawable.friedchicken to R.string.pack,  
    R.drawable.pasta to R.string.pasta,  
    R.drawable.gril to R.string.Chicken,  
    R.drawable.coco to R.string.wine,  
    R.drawable.deo to R.string.salad,  
    R.drawable.pizza to R.string.pizza,  
    R.drawable.momos to R.string.momos,  
    R.drawable.tandoori to R.string.tandoori,  
    R.drawable.friedchicken to R.string.friedchicken  
).map { DrawableStringPair(it.first, it.second) }
```

```
private data class DrawableStringPair(  
    @DrawableRes val drawable: Int,  
    @StringRes val text1: Int  
)
```

@Composable

```
fun App(context: Context) {
```

```
    Column(
```

```
        modifier = Modifier
```

```
            .fillMaxSize()
```

```
            .background(Color(0xffeceef0))
```

```
            .padding(10.dp),
```

```
        verticalArrangement = Arrangement.Top,
```

```
        horizontalAlignment = Alignment.CenterHorizontally
```

```
    ) {
```

```
        Surface(modifier = Modifier, elevation = 5.dp) {
```

```
            TopPart()
```

```
        }
```

```
        Spacer(modifier = Modifier.padding(10.dp))
```

```
        CardPart()
```

```
        Spacer(modifier = Modifier.padding(10.dp))
```

```
        Row(modifier = Modifier.fillMaxWidth(), Arrangement.SpaceBetween) {
```

```
            Text(text = "Popular Food", style = MaterialTheme.typography.h5, color = Color.Black)
```

```
            Text(text = "view all", style = MaterialTheme.typography.subtitle1, color = Color.Black)
```

```
        }
```

```
        Spacer(modifier = Modifier.padding(10.dp))
```

```
        PopularFoodColumn(context) // <- call the function with parentheses
```

```
    }
```

```
}
```

```
@Composable
fun PopularFoodColumn(context: Context) {
```

```
    LazyColumn(
        modifier = Modifier.fillMaxSize(),

        content = {
            items(FoodList) { item ->
                PopularFood(context = context, drawable = item.drawable, text1 = item.text1)
            }
        },
        verticalArrangement = Arrangement.spacedBy(16.dp))
}
```

```
@SuppressLint("UnusedMaterialScaffoldPaddingParameter")
```

```
@Composable
fun FinalView(mainPage: MainPage) {
    SnackOrderingTheme {
        Scaffold() {
            val context = LocalContext.current
            App(context)
        }
    }
}
```

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@drawable/fast_food"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/Theme.SnackOrdering"
        tools:targetApi="31">
        <activity
            android:name=".AdminActivity"
            android:exported="false"
            android:label="@string/title_activity_admin"
            android:theme="@style/Theme.SnackOrdering" />
```

```
<activity
    android:name=".LoginActivity"
    android:exported="true"
    android:label="SnackSquad"
    android:theme="@style/Theme.SnackOrdering">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name=".TargetActivity"
    android:exported="false"
    android:label="@string/title_activity_target"
    android:theme="@style/Theme.SnackOrdering" />
```



```
<activity
    android:name=".MainPage"
    android:exported="false"
    android:label="@string/title_activity_main_page"
    android:theme="@style/Theme.SnackOrdering" />
<activity
    android:name=".MainActivity"
    android:exported="false"
    android:label="MainActivity"
    android:theme="@style/Theme.SnackOrdering" />
</application>

</manifest>
```

OUTPUT:



