Express Eats (Snack Squad)

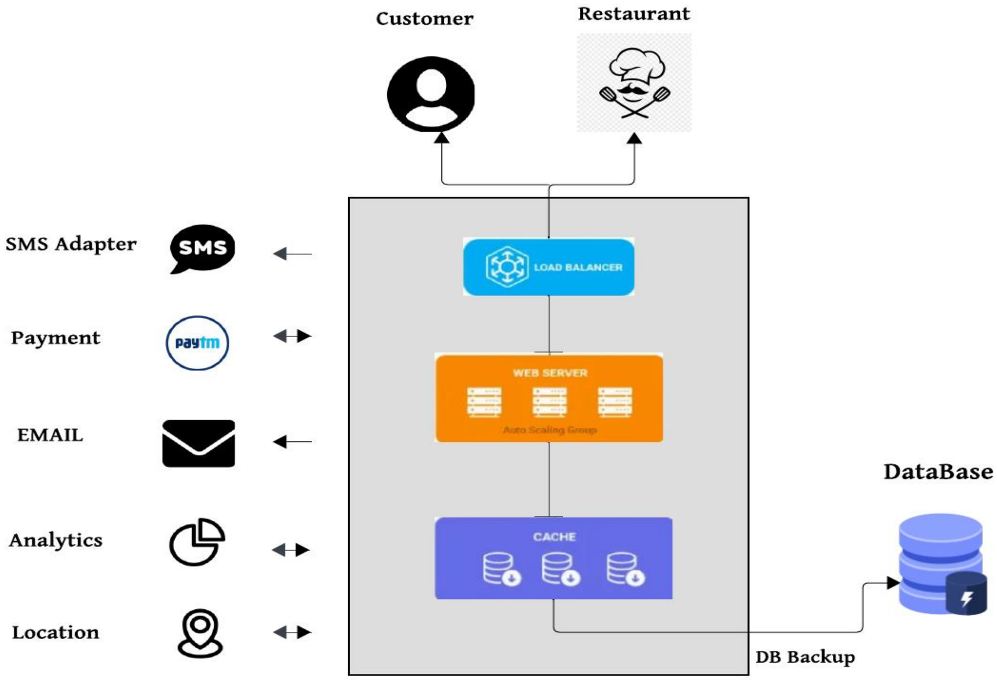
Introduction:

Food ordering apps have become essential for modern life due to their convenience and efficiency. They offer a hassle-free way to access a wide range of restaurant options, browse menus, and place orders at the touch of a button. With features like order tracking, payment options, and customization, these apps save time and provide a seamless dining experience, whether you're dining in or getting your favorite meals delivered to your doorstep. Food ordering apps are a convenient solution for busy individuals seeking a quick and enjoyable way to satisfy their culinary cravings.

Express Eats is your go-to food ordering app, designed to streamline the process of ordering and enjoying your favorite dishes from a variety of restaurants. Whether you're too busy to cook, want to enjoy a restaurant's delicacies at home, or just seeking a convenient dining experience, Express Eats is here to make your culinary journey hassle-free and delightful.

With a transparent and competitive commission fee structure, we reduce the financial burden on our restaurant partners. Our coupon system offers savings for customers while driving more business to our partner restaurants. Secure payment options, with industry-standard security protocols, protect financial data. The order process is hassle-free, with clear confirmation and tracking, and customers can access their order history for easy reordering and tracking past transactions.

Technical Architecture:



Prerequisites:

To complete this project, you must require the following software's, concepts, and packages

Software:

a**. Android Studio**:

Android Studio is the official integrated development environment for Android app development, providing tools, emulators, and resources for building Android applications.

b. **Kotlin:**

Kotlin is a modern, concise, and versatile programming language for Android development, known for its enhanced readability and maintainability compared to Java.

Concepts:

a. **RESTful API (Representational State Transfer):**

RESTful APIs are a set of architectural constraints used for designing networked applications. They enable communication between your Android app and a web server, supporting data retrieval and manipulation.

b. **Load Balancing:**

Load balancing distributes network traffic across multiple servers to ensure high availability and improved performance, making your app more resilient to server failures.

c**. Web Servers:**

A web server is a software that serves web content to users over the internet. It handles HTTP requests, processes them, and delivers web pages, allowing your app to interact with backend services.

Packages:

- **Material Components for Android:**
  Material Components is a design system for Android that offers ready-made UI components and design guidelines to ensure a consistent and visually appealing user interface in your app.

- **ViewModel and LiveData (from Android Architecture Components):**
  ViewModel and LiveData are part of Android Architecture Components and are vital for maintaining a separation of concerns and managing UI-related data in your app.

- **Hilt:**
  Hilt is a dependency injection library for Android that simplifies the process of injecting dependencies into your app's components, making it easier to manage dependencies and maintain clean code.

- **Room Persistence Library:**
  Room is an Android Architecture Component that provides an abstraction layer over SQLite. It simplifies database operations in your app and is an excellent choice for local

data storage.

- **Firebase (Realtime Database, Authentication, Cloud Functions):** Firebase is a comprehensive platform for mobile and web app development. It offers various services, including a Realtime Database, Authentication, and Cloud Functions, which can be used to handle data, user management, and serverless functions.

Project Objectives:

1. Enhance User Convenience
   Simplify menu browsing, order customization, and secure payments for a seamless user experience.

2. Expand Restaurant Selection
   Offer a diverse range of restaurants, both for delivery and dining in options.

3. Ensure Reliability
   Implement order tracking, efficient payment processing, and consistent service quality.

4. Optimize Performance
   Build a scalable system with load balancing to accommodate growing user demand and maintain efficiency.

Project Flow:

• **Sign-in/Sign-up:** Users start by signing in or creating an account if they're new to the app.

• **Restaurant Selection:** After signing in, users choose between delivery or dining in options.

• **Order Customization:** Users browse the selected restaurant's menu, add items to their cart, and make any desired customizations.

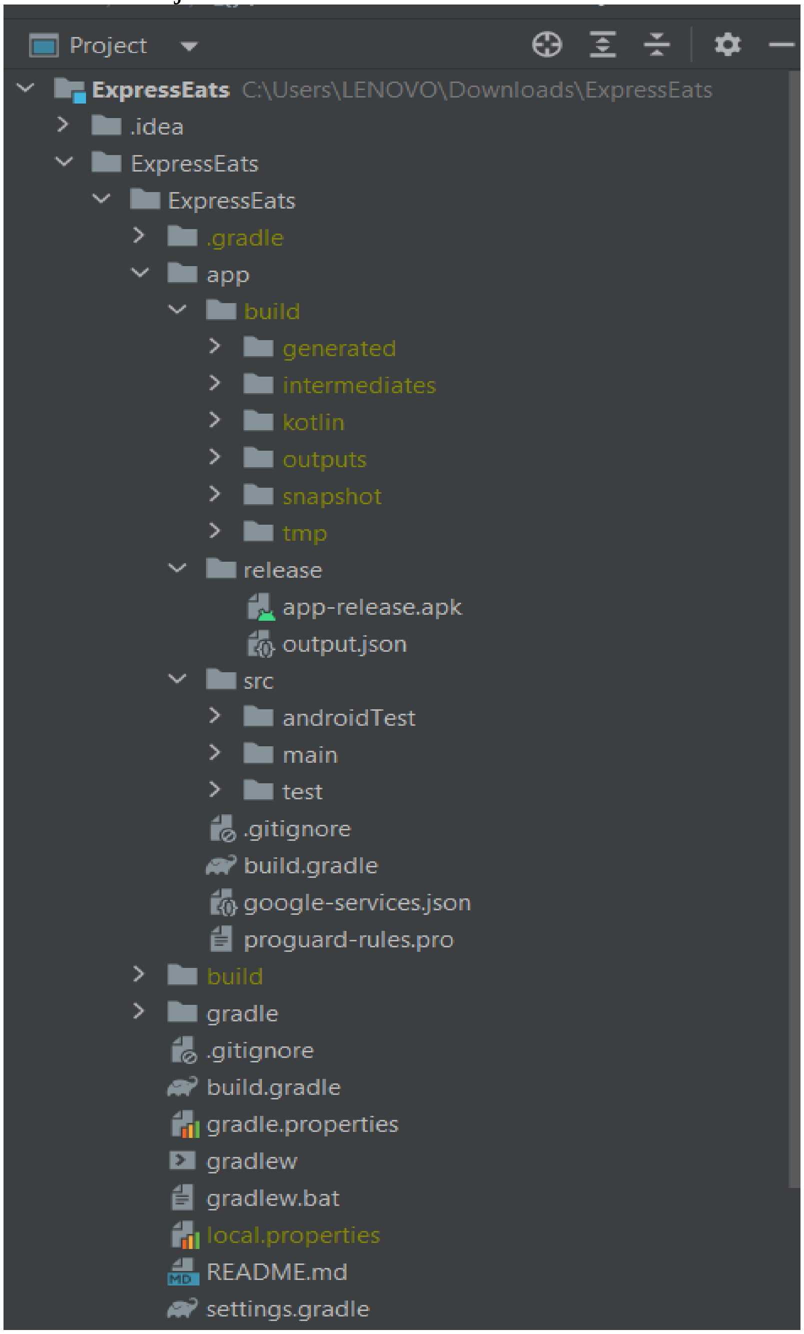
• **Payment Process:** Users proceed to payment, apply any available coupons, and enter their payment information.

• **Order Confirmation:** After confirming the order, the app processes the payment, and users receive a confirmation.

• **Order Tracking**: Users can view their order history and track the status of their current order in the "View My Orders" section.

Project Structure:

Create a Project folder which contains files as shown below

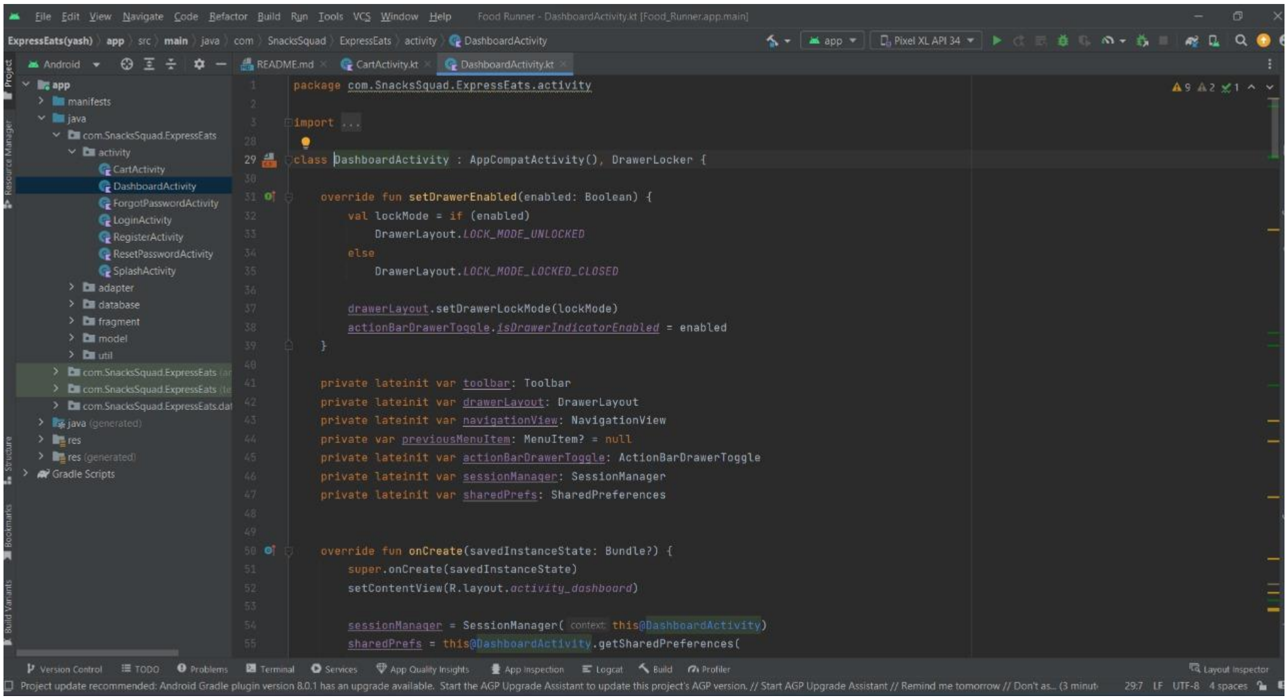


```
ExpressEats C:\Users\LENOVO\Downloads\ExpressEats
  .idea
  ExpressEats
    ExpressEats
      .gradle
      app
        build
          generated
          intermediates
          kotlin
          outputs
          snapshot
          tmp
        release
          app-release.apk
          output.json
        src
          androidTest
          main
          test
        .gitignore
        build.gradle
        google-services.json
        proguard-rules.pro
      build
      gradle
      .gitignore
      build.gradle
      gradle.properties
      gradlew
      gradlew.bat
      local.properties
      README.md
      settings.gradle
```

- **src:** This directory contains the source code for your app. It's further divided into subdirectories, such as:
- main: This is where most of your app's code resides.
- java: Java/Kotlin source code files for your app.
- res: Resources for your app, including XML layout files, drawable, strings, and other assets.
- test: Unit tests for your app.
- build. Gradle: This is the build configuration file for your app. It contains dependencies, build settings, and other project-level configurations.

- AndroidManifest.xml: This XML file defines essential information about your app, like its name, package name, permissions, and declared activities.

➢ **Dashboard Activity**

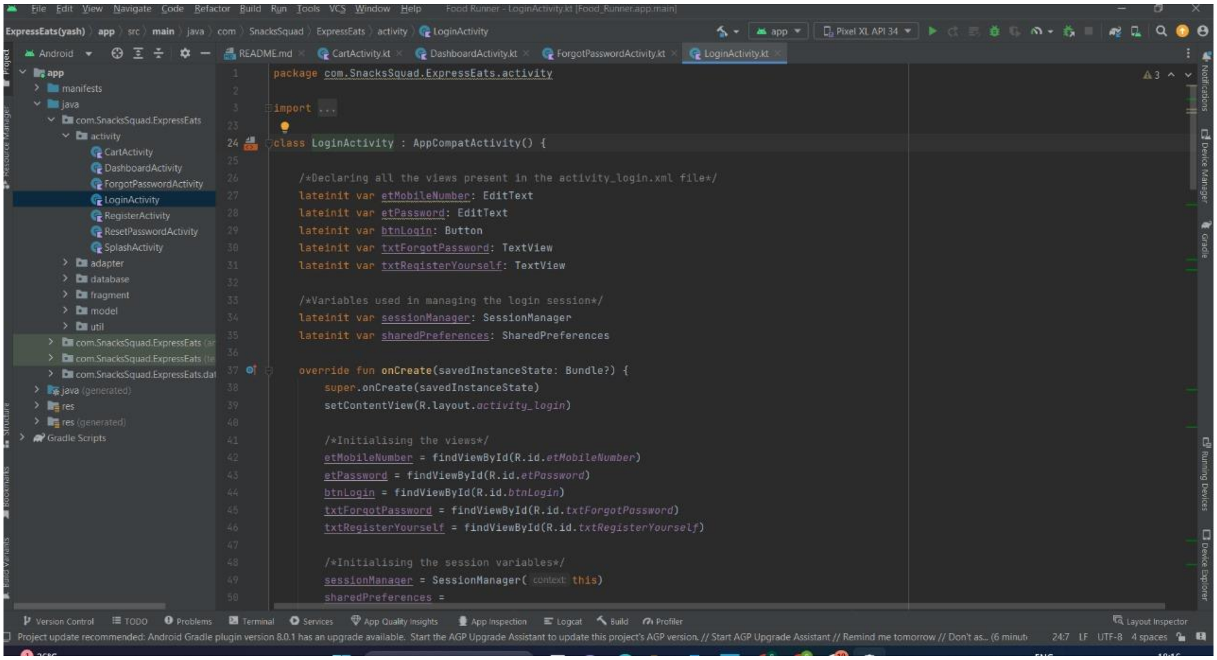


```kotlin
package com.SnacksSquad.ExpressEats.activity

import ...

class DashboardActivity : AppCompatActivity(), DrawerLocker {

    override fun setDrawerEnabled(enabled: Boolean) {
        val lockMode = if (enabled)
            DrawerLayout.LOCK_MODE_UNLOCKED
        else
            DrawerLayout.LOCK_MODE_LOCKED_CLOSED

        drawerLayout.setDrawerLockMode(lockMode)
        actionBarDrawerToggle.isDrawerIndicatorEnabled = enabled
    }

    private lateinit var toolbar: Toolbar
    private lateinit var drawerLayout: DrawerLayout
    private lateinit var navigationView: NavigationView
    private var previousMenuItem: MenuItem? = null
    private lateinit var actionBarDrawerToggle: ActionBarDrawerToggle
    private lateinit var sessionManager: SessionManager
    private lateinit var sharedPrefs: SharedPreferences


    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_dashboard)

        sessionManager = SessionManager( this@DashboardActivity)
        sharedPrefs = this@DashboardActivity.getSharedPreferences(
```

Project update recommended: Android Gradle plugin version 8.0.1 has an upgrade available. Start the AGP Upgrade Assistant to update this project's AGP version. // Start AGP Upgrade Assistant // Remind me tomorrow // Don't as... (3 minut 29:7 LF UTF-8 4 spaces

Once the restaurant owner sets up an account. They go back to the home page or dashboard. Here they can keep track of the pending orders, completed orders, and ones on the way to delivery for better insight and summary of what is going on.

➢ **Login**

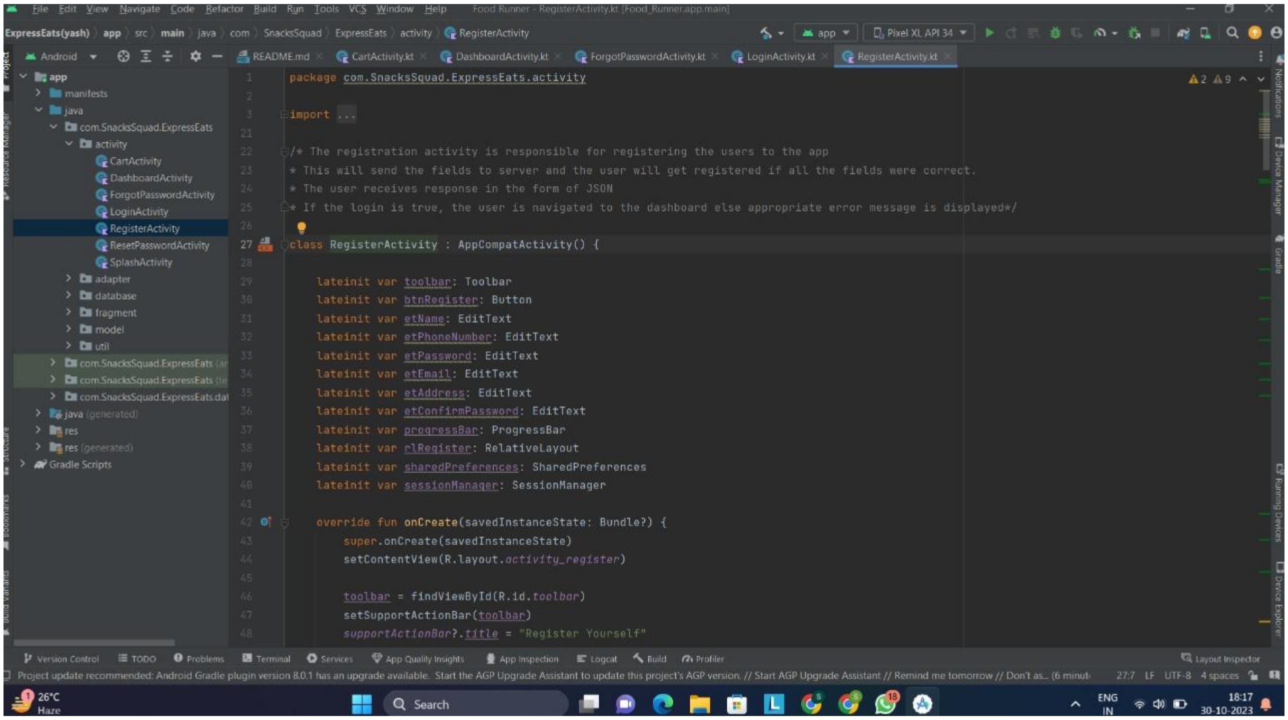


```kotlin
package com.SnacksSquad.ExpressEats.activity

import ...

class LoginActivity : AppCompatActivity() {

    /*Declaring all the views present in the activity_login.xml file*/
    lateinit var etMobileNumber: EditText
    lateinit var etPassword: EditText
    lateinit var btnLogin: Button
    lateinit var txtForgotPassword: TextView
    lateinit var txtRegisterYourself: TextView

    /*Variables used in managing the login session*/
    lateinit var sessionManager: SessionManager
    lateinit var sharedPreferences: SharedPreferences

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        /*Initialising the views*/
        etMobileNumber = findViewById(R.id.etMobileNumber)
        etPassword = findViewById(R.id.etPassword)
        btnLogin = findViewById(R.id.btnLogin)
        txtForgotPassword = findViewById(R.id.txtForgotPassword)
        txtRegisterYourself = findViewById(R.id.txtRegisterYourself)

        /*Initialising the session variables*/
        sessionManager = SessionManager( this)
        sharedPreferences =
```

Project update recommended: Android Gradle plugin version 8.0.1 has an upgrade available. Start the AGP Upgrade Assistant to update this project's AGP version. // Start AGP Upgrade Assistant // Remind me tomorrow // Don't as... (6 minut 24:7 LF UTF-8 4 spaces

The rider will receive login credentials from the admin to enter the application.

➢ Register/Sign up

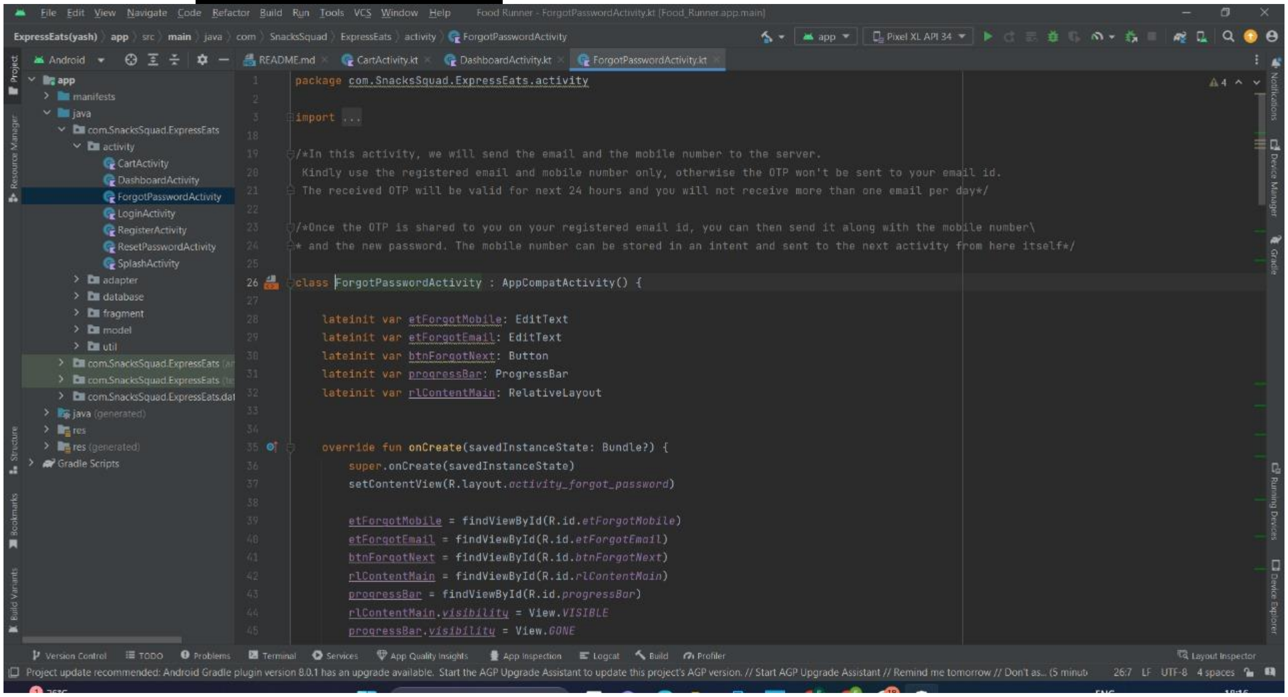


```
package com.SnacksSquad.ExpressEats.activity

import ...

/* The registration activity is responsible for registering the users to the app
* This will send the fields to server and the user will get registered if all the fields were correct.
* The user receives response in the form of JSON
* If the login is true, the user is navigated to the dashboard else appropriate error message is displayed*/

class RegisterActivity : AppCompatActivity() {

    lateinit var toolbar: Toolbar
    lateinit var btnRegister: Button
    lateinit var etName: EditText
    lateinit var etPhoneNumber: EditText
    lateinit var etPassword: EditText
    lateinit var etEmail: EditText
    lateinit var etAddress: EditText
    lateinit var etConfirmPassword: EditText
    lateinit var progressBar: ProgressBar
    lateinit var rlRegister: RelativeLayout
    lateinit var sharedPreferences: SharedPreferences
    lateinit var sessionManager: SessionManager

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_register)

        toolbar = findViewById(R.id.toolbar)
        setSupportActionBar(toolbar)
        supportActionBar?.title = "Register Yourself"
```

Project update recommended: Android Gradle plugin version 8.0.1 has an upgrade available. Start the AGP Upgrade Assistant to update this project's AGP version. // Start AGP Upgrade Assistant // Remind me tomorrow // Don't as... (6 minut 27:7 LF UTF-8 4 spaces

The page is about the sign-up process for a food delivery app. Users visit this page to create an account and gain access to the app's features, such as ordering food, tracking deliveries, and managing their profile. Users typically provide their personal information, including name, email, and password, to register on the platform.

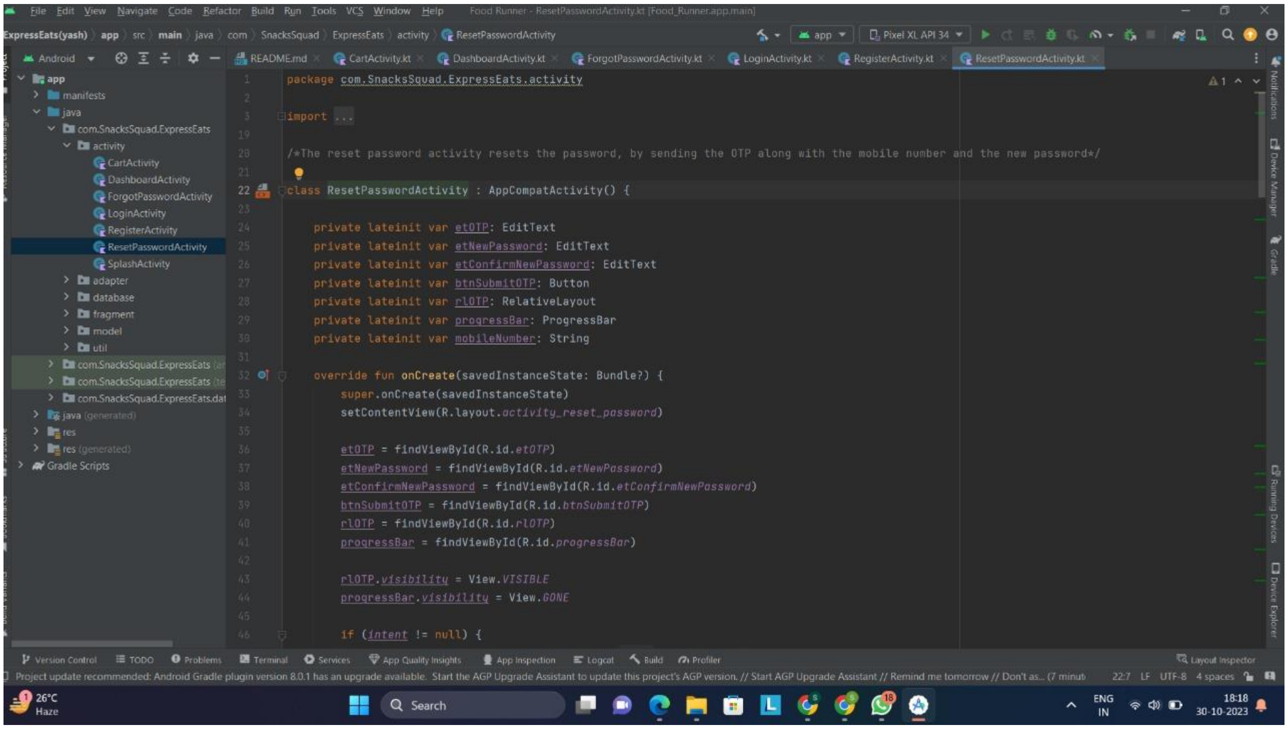
➢ Forget Password



```
package com.SnacksSquad.ExpressEats.activity

import ...

/*In this activity, we will send the email and the mobile number to the server.
 Kindly use the registered email and mobile number only, otherwise the OTP won't be sent to your email id.
 The received OTP will be valid for next 24 hours and you will not receive more than one email per day*/

/*Once the OTP is shared to you on your registered email id, you can then send it along with the mobile number\
* and the new password. The mobile number can be stored in an intent and sent to the next activity from here itself*/

class ForgotPasswordActivity : AppCompatActivity() {

    lateinit var etForgotMobile: EditText
    lateinit var etForgotEmail: EditText
    lateinit var btnForgotNext: Button
    lateinit var progressBar: ProgressBar
    lateinit var rlContentMain: RelativeLayout


    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_forgot_password)

        etForgotMobile = findViewById(R.id.etForgotMobile)
        etForgotEmail = findViewById(R.id.etForgotEmail)
        btnForgotNext = findViewById(R.id.btnForgotNext)
        rlContentMain = findViewById(R.id.rlContentMain)
        progressBar = findViewById(R.id.progressBar)
        rlContentMain.visibility = View.VISIBLE
        progressBar.visibility = View.GONE
```

Project update recommended: Android Gradle plugin version 8.0.1 has an upgrade available. Start the AGP Upgrade Assistant to update this project's AGP version. // Start AGP Upgrade Assistant // Remind me tomorrow // Don't as... (5 minut 26:7 LF UTF-8 4 spaces

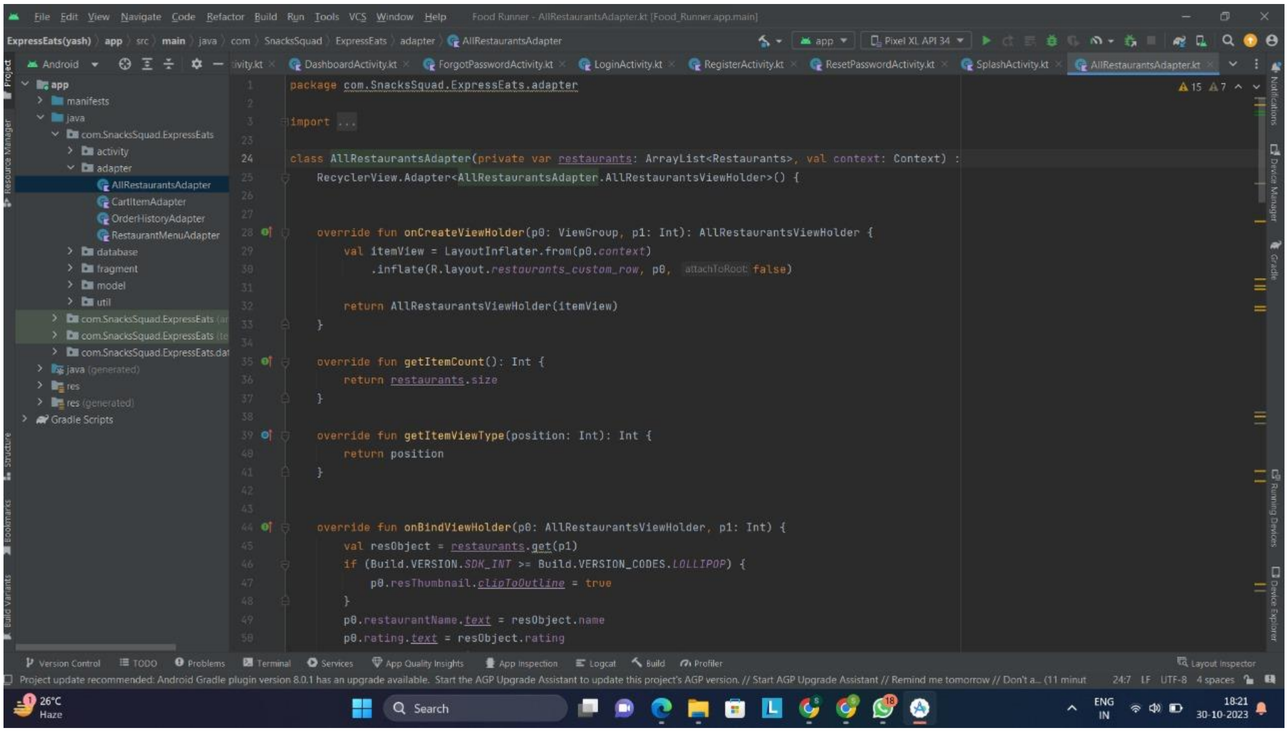
➢ Reset Password



```
package com.SnacksSquad.ExpressEats.activity

import ...

/*The reset password activity resets the password, by sending the OTP along with the mobile number and the new password*/

class ResetPasswordActivity : AppCompatActivity() {

    private lateinit var etOTP: EditText
    private lateinit var etNewPassword: EditText
    private lateinit var etConfirmNewPassword: EditText
    private lateinit var btnSubmitOTP: Button
    private lateinit var rlOTP: RelativeLayout
    private lateinit var progressBar: ProgressBar
    private lateinit var mobileNumber: String

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_reset_password)

        etOTP = findViewById(R.id.etOTP)
        etNewPassword = findViewById(R.id.etNewPassword)
        etConfirmNewPassword = findViewById(R.id.etConfirmNewPassword)
        btnSubmitOTP = findViewById(R.id.btnSubmitOTP)
        rlOTP = findViewById(R.id.rlOTP)
        progressBar = findViewById(R.id.progressBar)

        rlOTP.visibility = View.VISIBLE
        progressBar.visibility = View.GONE

        if (intent != null) {
```

Project update recommended: Android Gradle plugin version 8.0.1 has an upgrade available. Start the AGP Upgrade Assistant to update this project's AGP version. // Start AGP Upgrade Assistant // Remind me tomorrow // Don't as... (7 minut

➢ Restaurant List

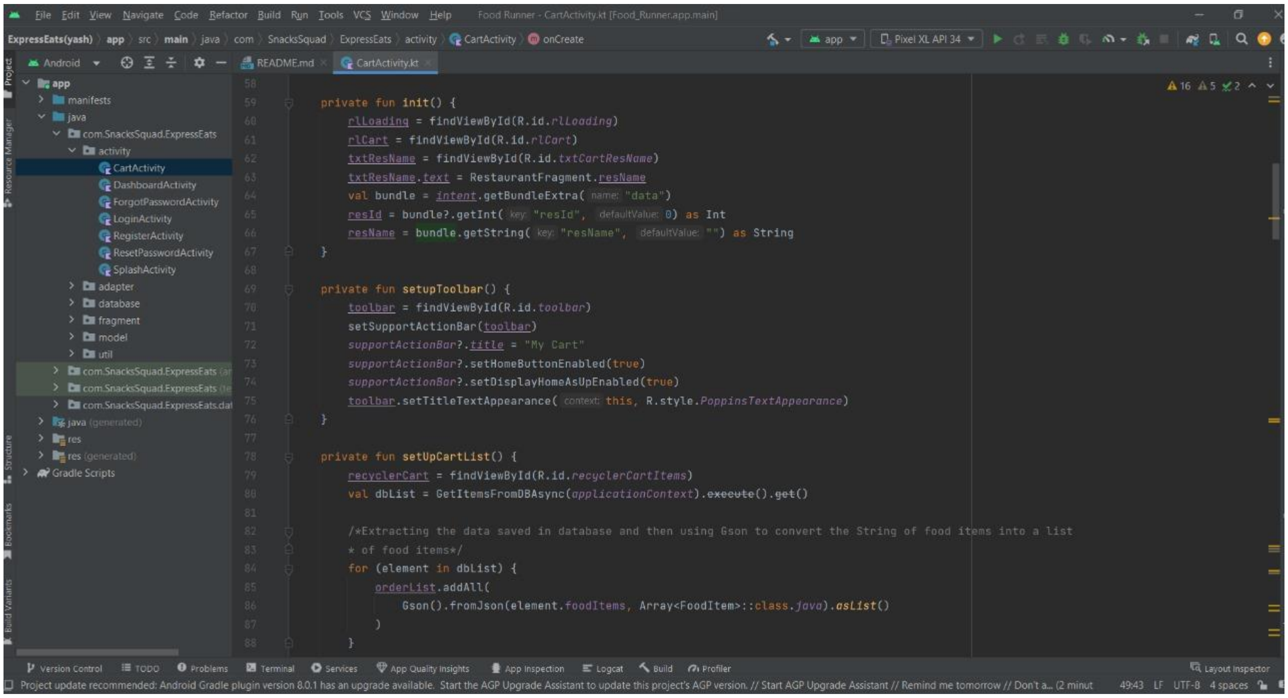


```
package com.SnacksSquad.ExpressEats.adapter

import ...

class AllRestaurantsAdapter(private var restaurants: ArrayList<Restaurants>, val context: Context) :
    RecyclerView.Adapter<AllRestaurantsAdapter.AllRestaurantsViewHolder>() {


    override fun onCreateViewHolder(p0: ViewGroup, p1: Int): AllRestaurantsViewHolder {
        val itemView = LayoutInflater.from(p0.context)
            .inflate(R.layout.restaurants_custom_row, p0,  attachToRoot: false)

        return AllRestaurantsViewHolder(itemView)
    }

    override fun getItemCount(): Int {
        return restaurants.size
    }

    override fun getItemViewType(position: Int): Int {
        return position
    }


    override fun onBindViewHolder(p0: AllRestaurantsViewHolder, p1: Int) {
        val resObject = restaurants.get(p1)
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            p0.resThumbnail.clipToOutline = true
        }
        p0.restaurantName.text = resObject.name
        p0.rating.text = resObject.rating
```

Project update recommended: Android Gradle plugin version 8.0.1 has an upgrade available. Start the AGP Upgrade Assistant to update this project's AGP version. // Start AGP Upgrade Assistant // Remind me tomorrow // Don't a... (11 minut

A restaurant list page on a food delivery app typically displays a curated or search-based list of available restaurants. Users can browse through these lists to view restaurant options, their menus, ratings, and reviews. This page allows users to select a restaurant they want to order from, making it a crucial step in the food ordering process.

➢ CART ACTIVITY

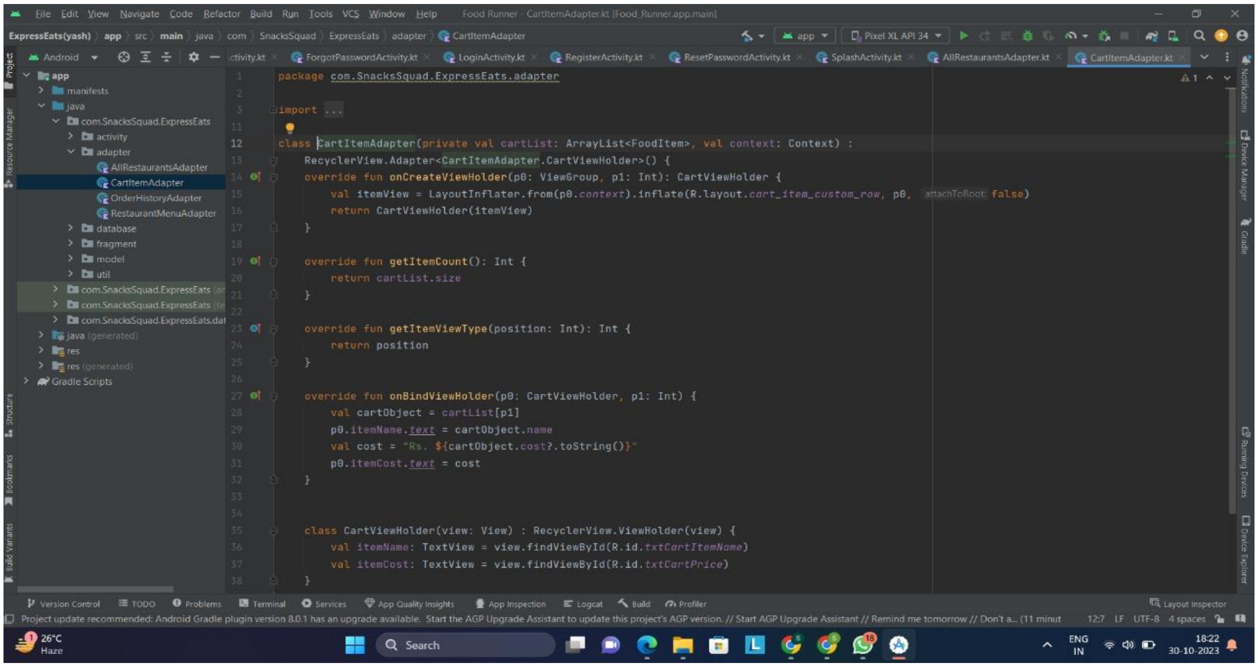


```
    private fun init() {
        rlLoading = findViewById(R.id.rlLoading)
        rlCart = findViewById(R.id.rlCart)
        txtResName = findViewById(R.id.txtCartResName)
        txtResName.text = RestaurantFragment.resName
        val bundle = intent.getBundleExtra( name: "data")
        resId = bundle?.getInt( key: "resId",  defaultValue: 0) as Int
        resName = bundle.getString( key: "resName",  defaultValue: "") as String
    }

    private fun setupToolbar() {
        toolbar = findViewById(R.id.toolbar)
        setSupportActionBar(toolbar)
        supportActionBar?.title = "My Cart"
        supportActionBar?.setHomeButtonEnabled(true)
        supportActionBar?.setDisplayHomeAsUpEnabled(true)
        toolbar.setTitleTextAppearance( context: this, R.style.PoppinsTextAppearance)
    }

    private fun setUpCartList() {
        recyclerCart = findViewById(R.id.recyclerCartItems)
        val dbList = GetItemsFromDBAsync(applicationContext).execute().get()

        /*Extracting the data saved in database and then using Gson to convert the String of food items into a list
        * of food items*/
        for (element in dbList) {
            orderList.addAll(
                Gson().fromJson(element.foodItems, Array<FoodItem>::class.java).asList()
            )
        }
```

Project update recommended: Android Gradle plugin version 8.0.1 has an upgrade available. Start the AGP Upgrade Assistant to update this project's AGP version. // Start AGP Upgrade Assistant // Remind me tomorrow // Don't a... (2 minut

A cart activity page in a food delivery app allows users to review and manage the items they've added to their order. Users can see the selected food items, their quantities, and prices, and make adjustments like adding or removing items. They can also proceed to the checkout and payment process from this page. It serves as a summary and control center for their current order.

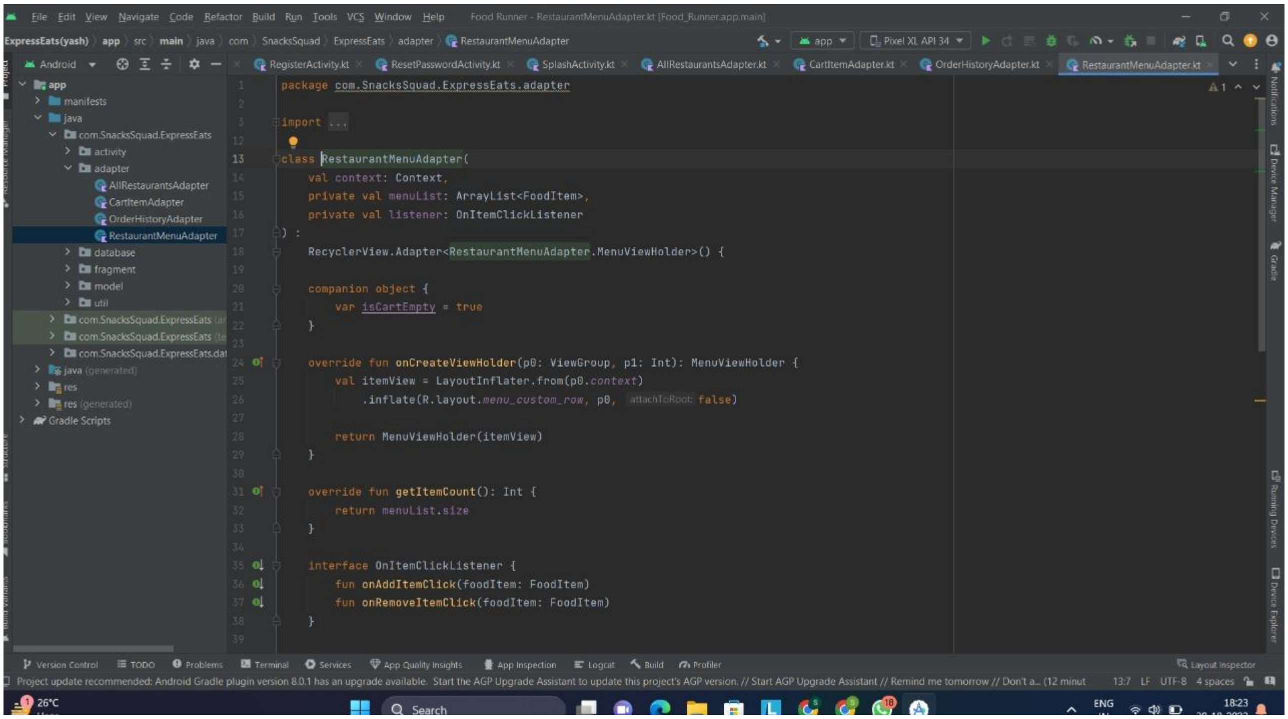
➢ Cart Item



```
package com.SnacksSquad.ExpressEats.adapter

import ...

class CartItemAdapter(private val cartList: ArrayList<FoodItem>, val context: Context) :
    RecyclerView.Adapter<CartItemAdapter.CartViewHolder>() {
    override fun onCreateViewHolder(p0: ViewGroup, p1: Int): CartViewHolder {
        val itemView = LayoutInflater.from(p0.context).inflate(R.layout.cart_item_custom_row, p0,  attachToRoot: false)
        return CartViewHolder(itemView)
    }

    override fun getItemCount(): Int {
        return cartList.size
    }

    override fun getItemViewType(position: Int): Int {
        return position
    }

    override fun onBindViewHolder(p0: CartViewHolder, p1: Int) {
        val cartObject = cartList[p1]
        p0.itemName.text = cartObject.name
        val cost = "Rs. ${cartObject.cost?.toString()}"
        p0.itemCost.text = cost
    }


    class CartViewHolder(view: View) : RecyclerView.ViewHolder(view) {
        val itemName: TextView = view.findViewById(R.id.txtCartItemName)
        val itemCost: TextView = view.findViewById(R.id.txtCartPrice)
    }
```

Project update recommended: Android Gradle plugin version 8.0.1 has an upgrade available. Start the AGP Upgrade Assistant to update this project's AGP version. // Start AGP Upgrade Assistant // Remind me tomorrow // Don't a... (11 minut

➢ Menu

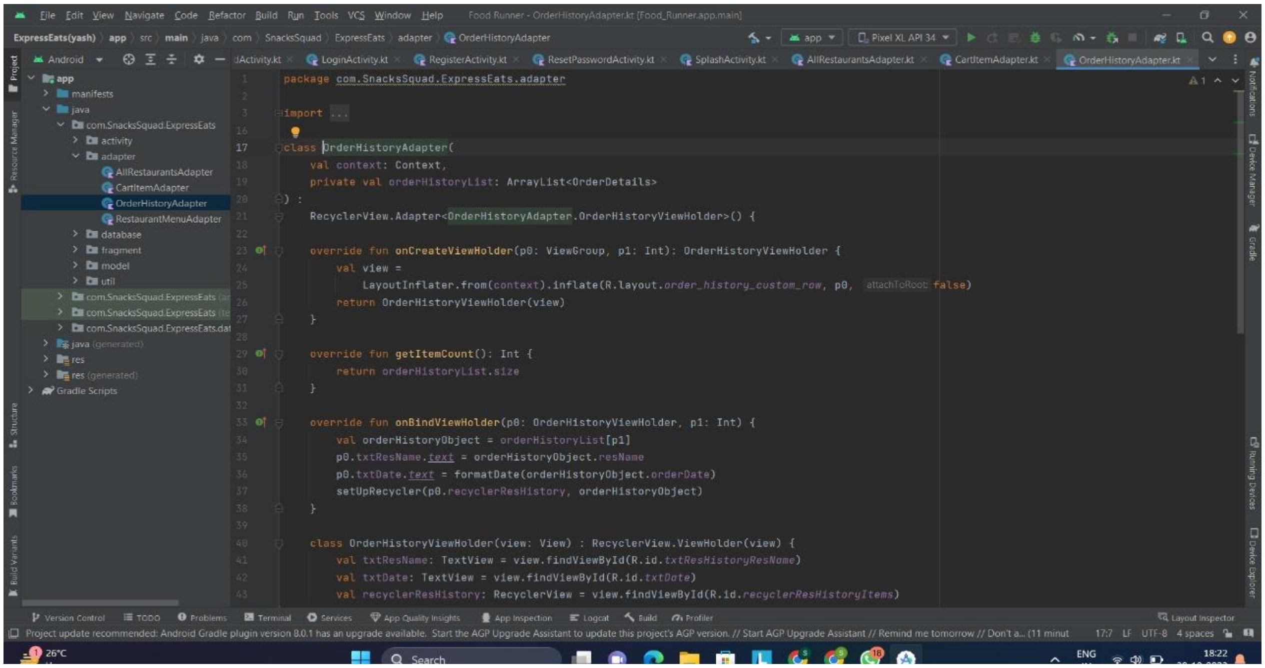


```kotlin
package com.SnacksSquad.ExpressEats.adapter

import ...

class RestaurantMenuAdapter(
    val context: Context,
    private val menuList: ArrayList<FoodItem>,
    private val listener: OnItemClickListener
) :
    RecyclerView.Adapter<RestaurantMenuAdapter.MenuViewHolder>() {

    companion object {
        var isCartEmpty = true
    }

    override fun onCreateViewHolder(p0: ViewGroup, p1: Int): MenuViewHolder {
        val itemView = LayoutInflater.from(p0.context)
            .inflate(R.layout.menu_custom_row, p0,  attachToRoot: false)

        return MenuViewHolder(itemView)
    }

    override fun getItemCount(): Int {
        return menuList.size
    }

    interface OnItemClickListener {
        fun onAddItemClick(foodItem: FoodItem)
        fun onRemoveItemClick(foodItem: FoodItem)
    }

```

```
Project update recommended: Android Gradle plugin version 8.0.1 has an upgrade available. Start the AGP Upgrade Assistant to update this project's AGP version. // Start AGP Upgrade Assistant // Remind me tomorrow // Don't a... (12 minut
```

A menu page in a food delivery app presents the list of dishes and items available from a specific restaurant. Users can browse through the menu to view food options, descriptions, prices, and sometimes images. They can select items they want to add to their order from this page, customizing their meal choices before proceeding to the cart or checkout.

➢ Order History

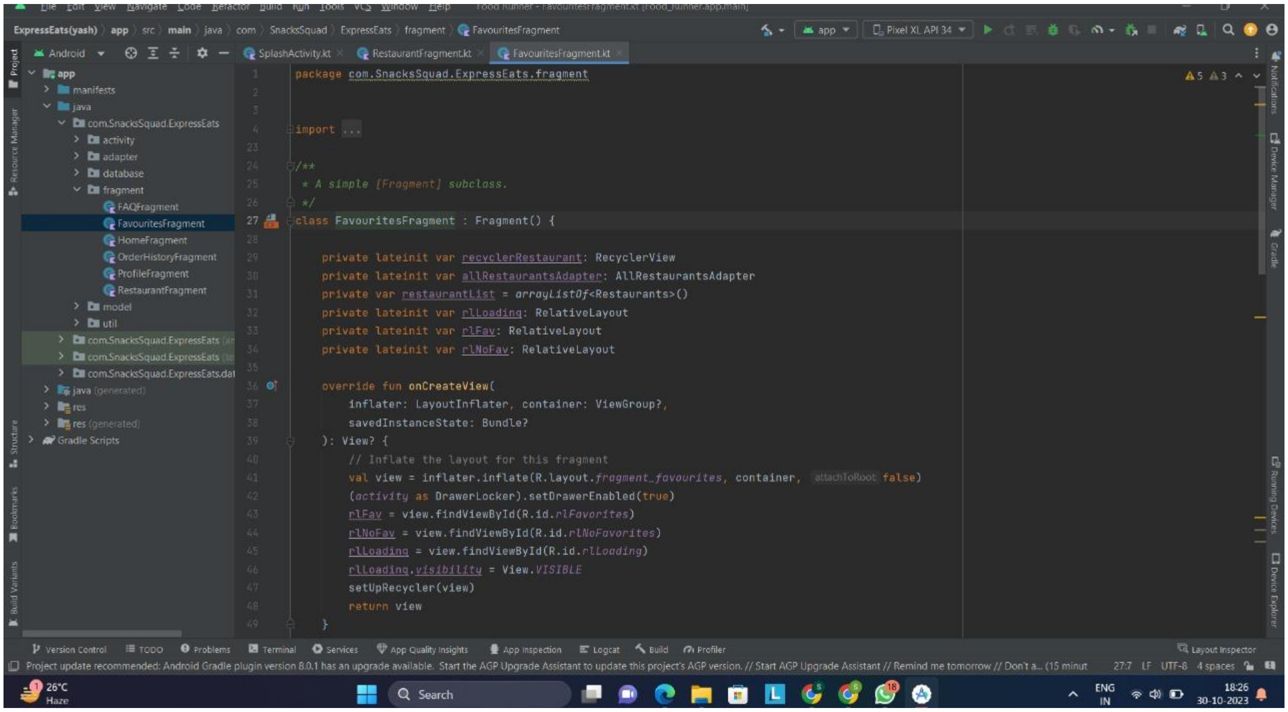


```kotlin
package com.SnacksSquad.ExpressEats.adapter

import ...

class OrderHistoryAdapter(
    val context: Context,
    private val orderHistoryList: ArrayList<OrderDetails>
) :
    RecyclerView.Adapter<OrderHistoryAdapter.OrderHistoryViewHolder>() {

    override fun onCreateViewHolder(p0: ViewGroup, p1: Int): OrderHistoryViewHolder {
        val view =
            LayoutInflater.from(context).inflate(R.layout.order_history_custom_row, p0,  attachToRoot: false)
        return OrderHistoryViewHolder(view)
    }

    override fun getItemCount(): Int {
        return orderHistoryList.size
    }

    override fun onBindViewHolder(p0: OrderHistoryViewHolder, p1: Int) {
        val orderHistoryObject = orderHistoryList[p1]
        p0.txtResName.text = orderHistoryObject.resName
        p0.txtDate.text = formatDate(orderHistoryObject.orderDate)
        setUpRecycler(p0.recyclerResHistory, orderHistoryObject)
    }

    class OrderHistoryViewHolder(view: View) : RecyclerView.ViewHolder(view) {
        val txtResName: TextView = view.findViewById(R.id.txtResHistoryResName)
        val txtDate: TextView = view.findViewById(R.id.txtDate)
        val recyclerResHistory: RecyclerView = view.findViewById(R.id.recyclerResHistoryItems)
```

```
Project update recommended: Android Gradle plugin version 8.0.1 has an upgrade available. Start the AGP Upgrade Assistant to update this project's AGP version. // Start AGP Upgrade Assistant // Remind me tomorrow // Don't a... (11 minut
```

An order history page in a food delivery app displays a record of all the orders a user has placed in the past. Users can review details of previous orders, such as the date, time, items ordered, total cost, and delivery status. It allows users to keep track of their ordering history and easily reorder their favorite meals or check the status of recent orders.

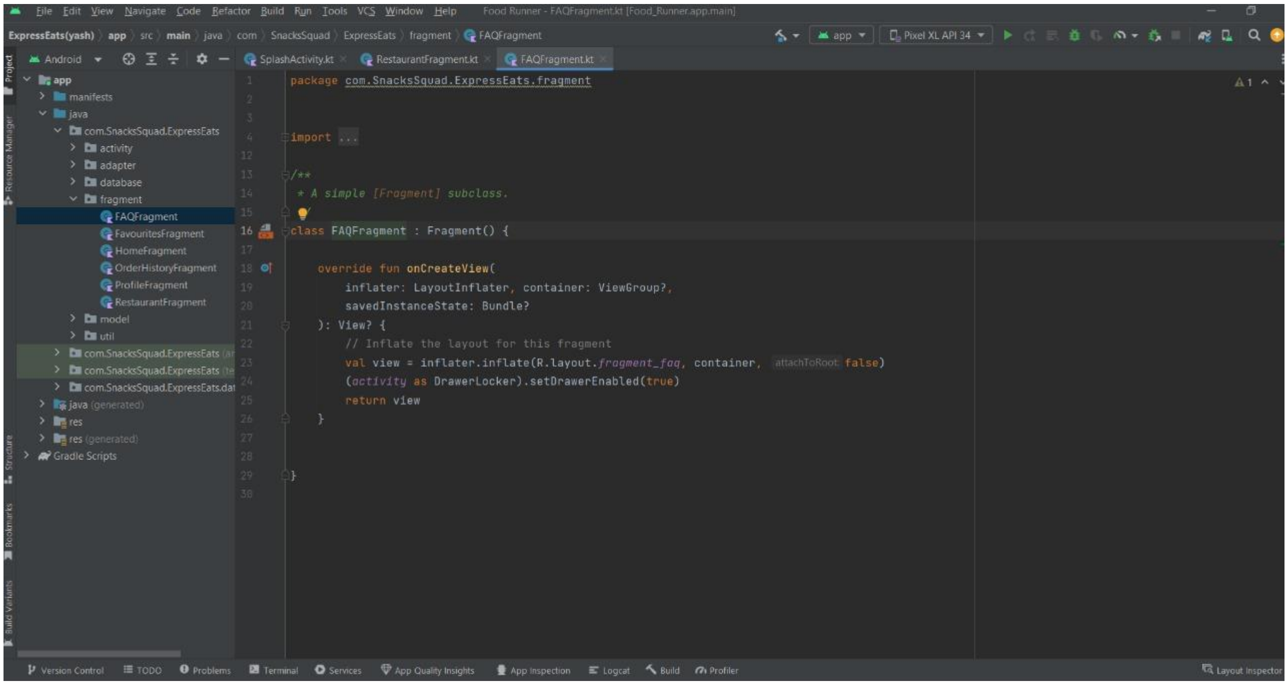
- ## Favorite



```kotlin
package com.SnacksSquad.ExpressEats.fragment


import ...

/**
 * A simple [Fragment] subclass.
 */
class FavouritesFragment : Fragment() {

    private lateinit var recyclerRestaurant: RecyclerView
    private lateinit var allRestaurantsAdapter: AllRestaurantsAdapter
    private var restaurantList = arrayListOf<Restaurants>()
    private lateinit var rlLoading: RelativeLayout
    private lateinit var rlFav: RelativeLayout
    private lateinit var rlNoFav: RelativeLayout

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        val view = inflater.inflate(R.layout.fragment_favourites, container,  attachToRoot: false)
        (activity as DrawerLocker).setDrawerEnabled(true)
        rlFav = view.findViewById(R.id.rlFavorites)
        rlNoFav = view.findViewById(R.id.rlNoFavorites)
        rlLoading = view.findViewById(R.id.rlLoading)
        rlLoading.visibility = View.VISIBLE
        setUpRecycler(view)
        return view
    }
```

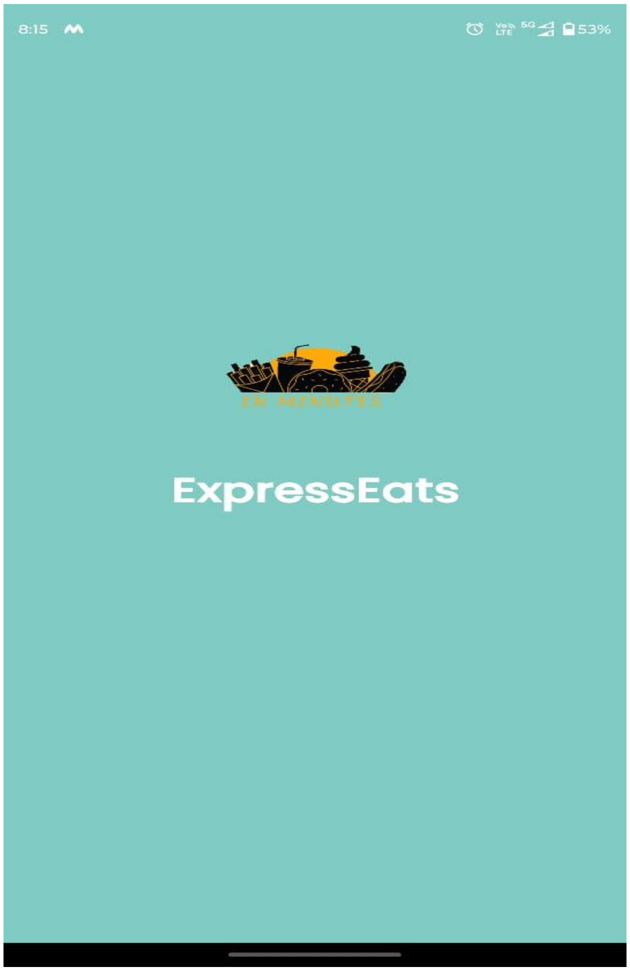
Project update recommended: Android Gradle plugin version 8.0.1 has an upgrade available. Start the AGP Upgrade Assistant to update this project's AGP version. // Start AGP Upgrade Assistant // Remind me tomorrow // Don't a... (15 minut

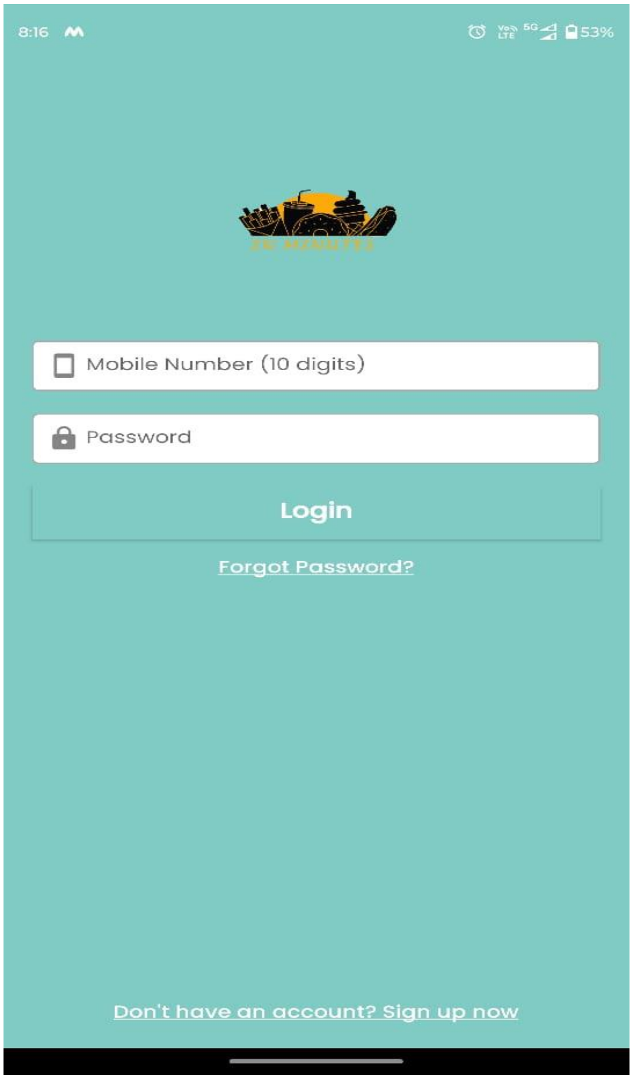
- FAQ

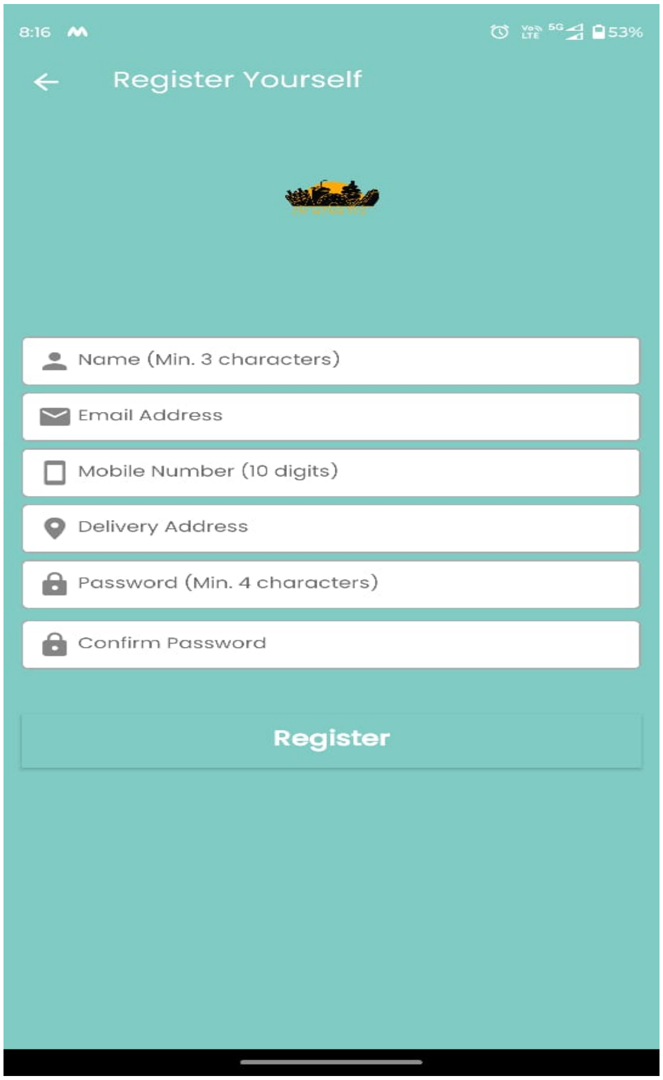

Food Runner - FAQFragment.kt [Food_Runner.app.main]
ExpressEats(yash) 〉 app 〉 src 〉 main 〉 java 〉 com 〉 SnacksSquad 〉 ExpressEats 〉 fragment 〉 FAQFragment
SplashActivity.kt
RestaurantFragment.kt
FAQFragment.kt

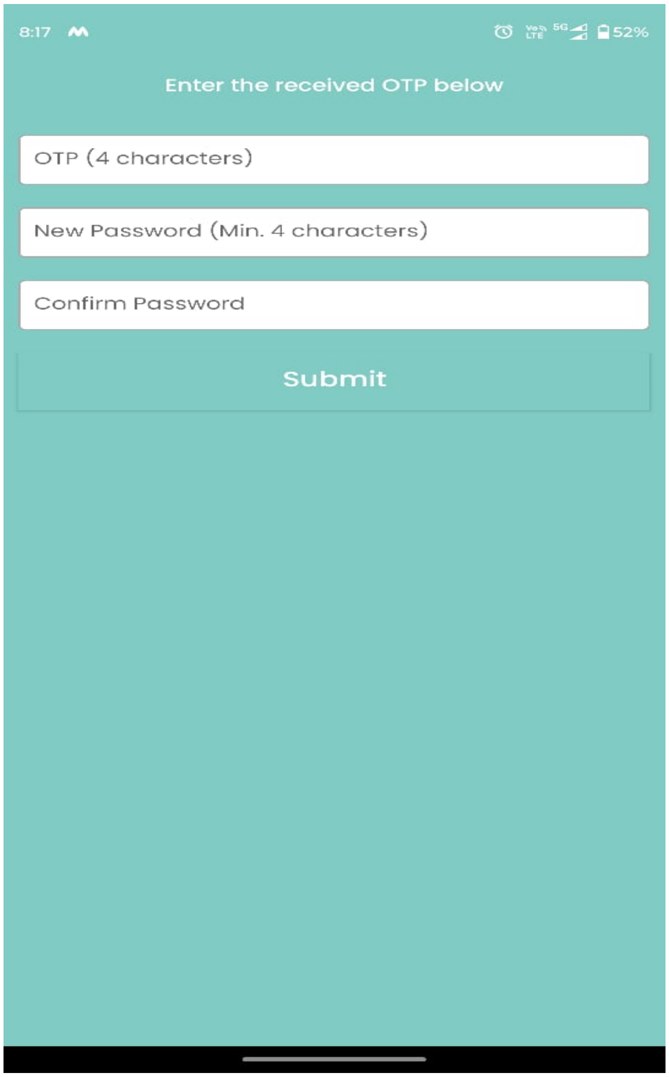
```kotlin
package com.SnacksSquad.ExpressEats.fragment


import ...

/**
 * A simple [Fragment] subclass.

class FAQFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        val view = inflater.inflate(R.layout.fragment_faq, container,  attachToRoot: false)
        (activity as DrawerLocker).setDrawerEnabled(true)
        return view
    }


}
```

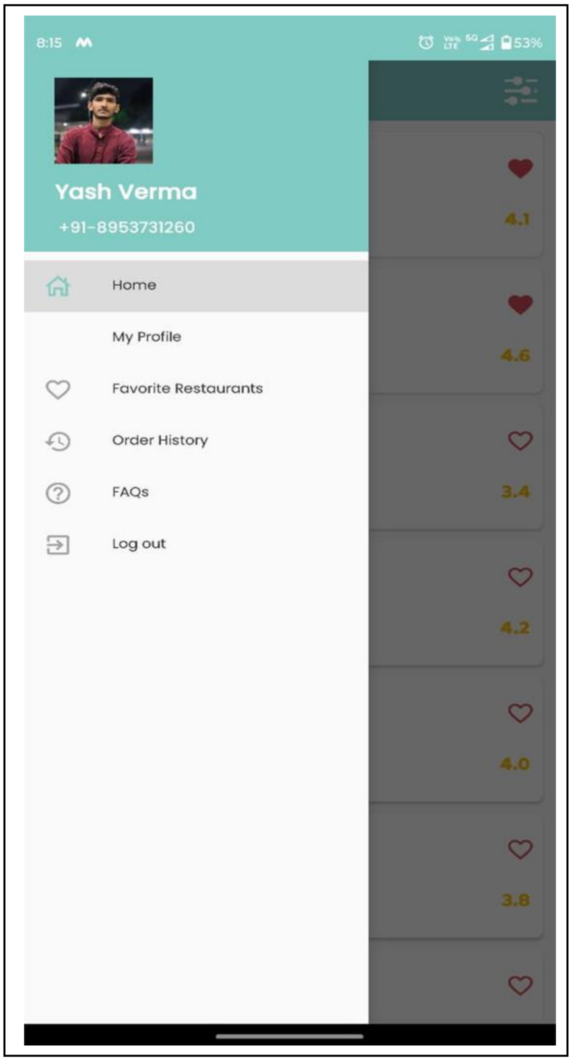
APP

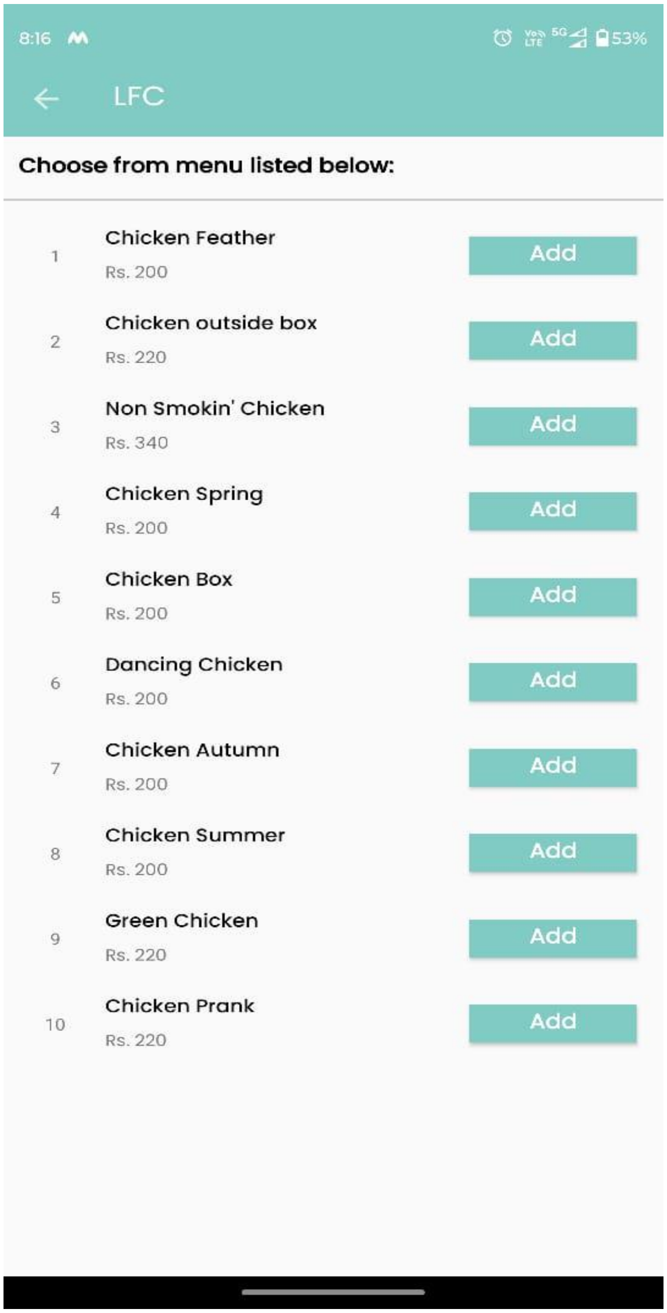



8:16
53%
Mobile Number (10 digits)
Password
Login
Forgot Password?
Don't have an account? Sign up now


8:16
53%
Register Yourself
Name (Min. 3 characters)
Email Address
Mobile Number (10 digits)
Delivery Address
Password (Min. 4 characters)
Confirm Password
Register

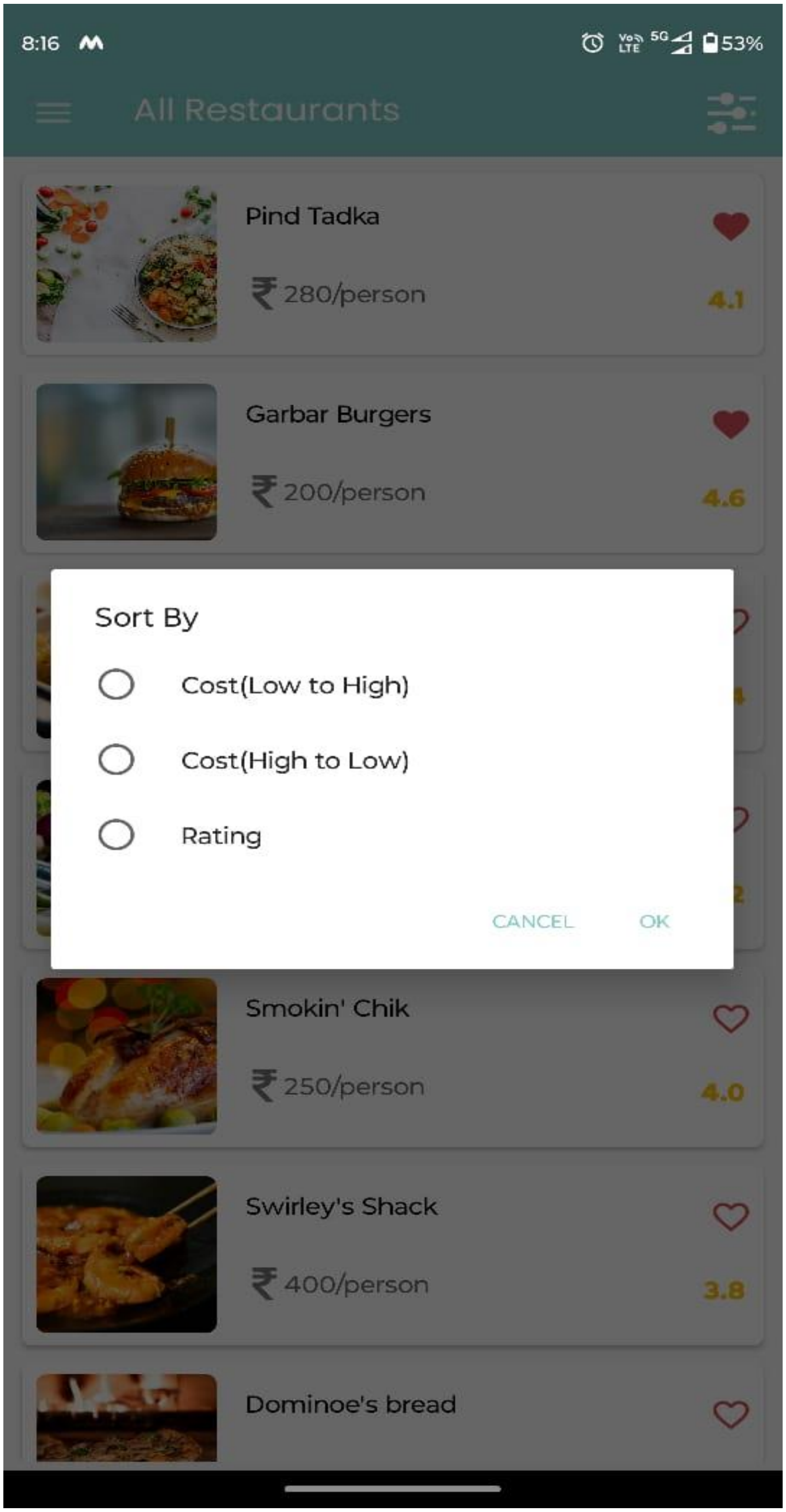

Enter the received OTP below





Choose from menu listed below:

1	Chicken Feather	Rs. 200
2	Chicken outside box	Rs. 220
3	Non Smokin' Chicken	Rs. 340
4	Chicken Spring	Rs. 200
5	Chicken Box	Rs. 200
6	Dancing Chicken	Rs. 200
7	Chicken Autumn	Rs. 200
8	Chicken Summer	Rs. 200
9	Green Chicken	Rs. 220
10	Chicken Prank	Rs. 220



Pind Tadka
₹ 280/person
4.1

Garbar Burgers
₹ 200/person
4.6

Sort By

○ Cost(Low to High)

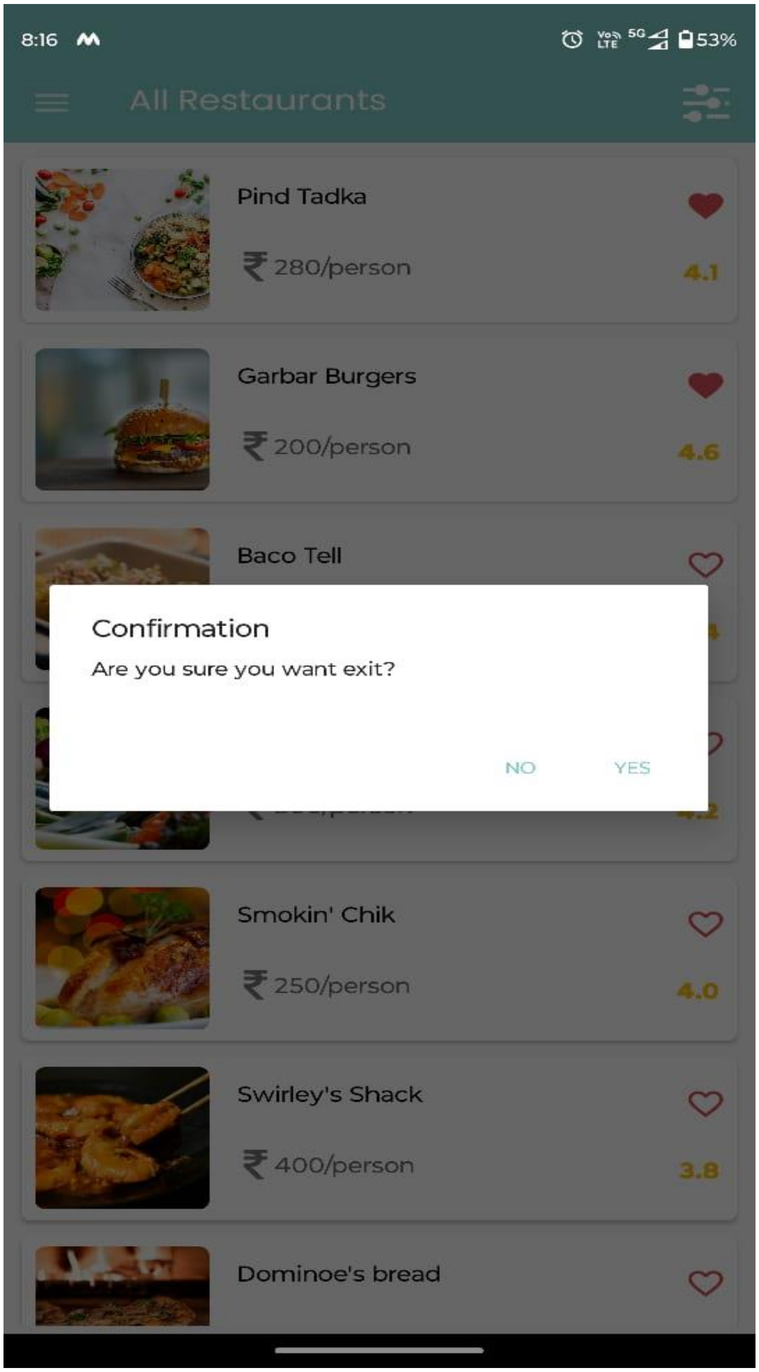
○ Cost(High to Low)

○ Rating

Smokin' Chik
₹ 250/person
4.0

Swirley's Shack
₹ 400/person
3.8

Dominoe's bread



All Restaurants

Pind Tadka
₹280/person
4.1

Garbar Burgers
₹200/person
4.6

Baco Tell

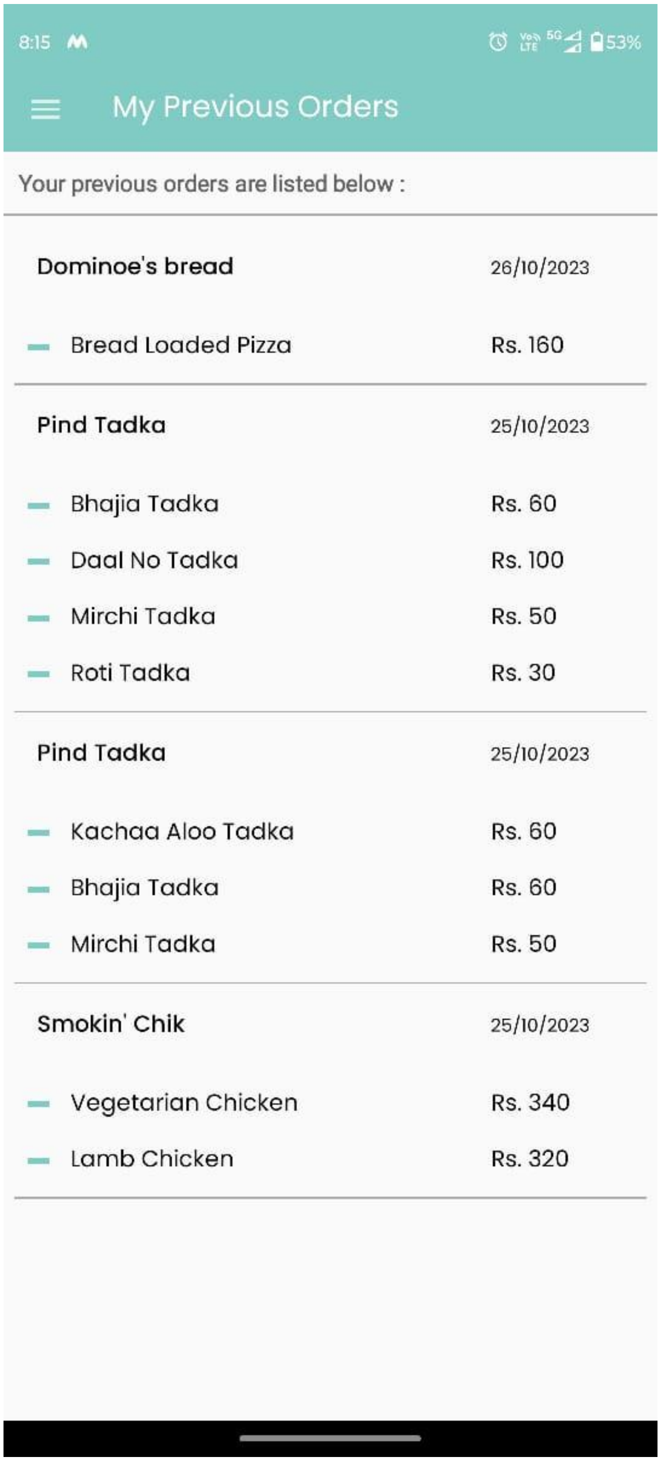
Confirmation

Are you sure you want exit?

Smokin' Chik
₹250/person
4.0

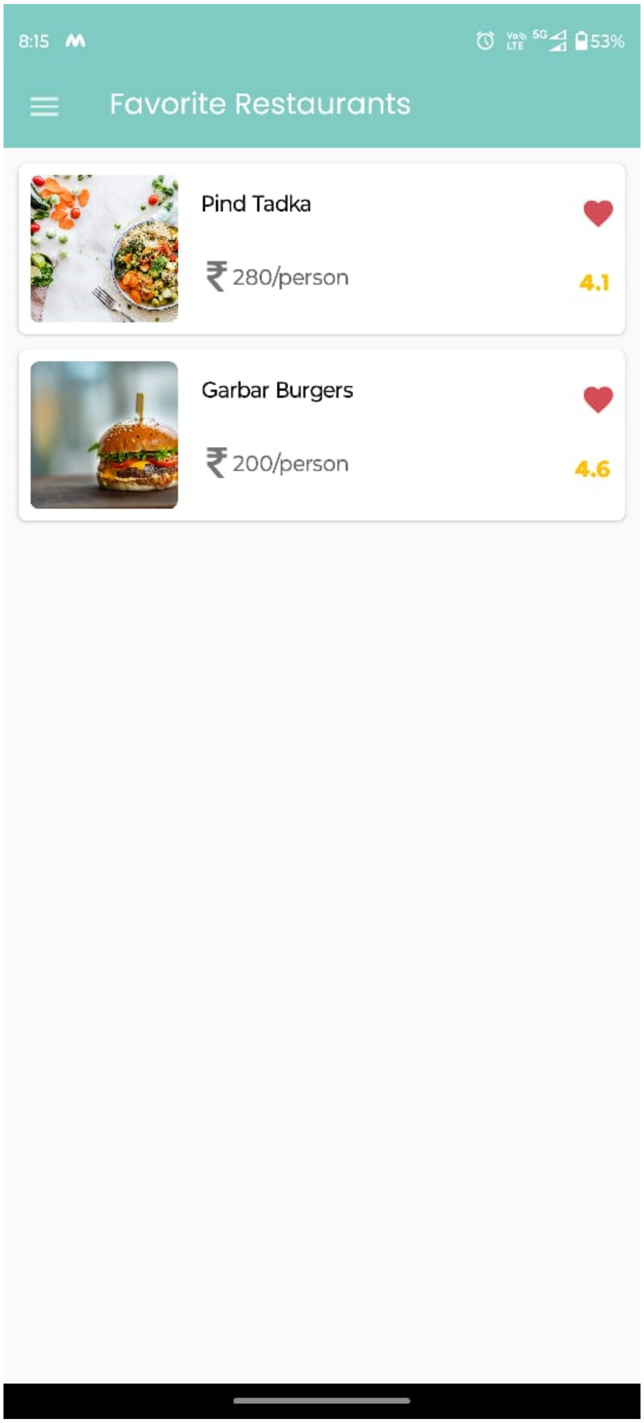
Swirley's Shack
₹400/person
3.8

Dominoe's bread



Your previous orders are listed below :

Dominoe's bread	26/10/2023
Bread Loaded Pizza	Rs. 160
Pind Tadka	25/10/2023
Bhajia Tadka	Rs. 60
Daal No Tadka	Rs. 100
Mirchi Tadka	Rs. 50
Roti Tadka	Rs. 30
Pind Tadka	25/10/2023
Kachaa Aloo Tadka	Rs. 60
Bhajia Tadka	Rs. 60
Mirchi Tadka	Rs. 50
Smokin' Chik	25/10/2023
Vegetarian Chicken	Rs. 340
Lamb Chicken	Rs. 320



Favorite Restaurants

Pind Tadka

₹ 280/person

4.1

Garbar Burgers

₹ 200/person

4.6