**Report on Android Food Ordering App**

**Introduction**

The project is an Android-based food ordering application that allows users to order various food items from a predefined menu. The app consists of two main activities: the **MainActivity**, where the user can see the ordered items and their quantities, and the **MenuActivity**, where users can select items like Hamburger, Pizza, Ice Cream, Coke, and Chips. The app leverages a RecyclerView with an Adapter to display the list of ordered items in a structured manner, ensuring a scalable and user-friendly interface.

**Features Implemented**

1. **Menu with Food Items**: The app features a menu in the MenuActivity, where users can add items to their order. Each food item is displayed with its name, price, and a button for adding it to the order.
2. **Order List**: In MainActivity, users can view the items they've added to their order. The app consolidates repeated orders by updating the quantity rather than listing the same item multiple times. For example, ordering two hamburgers shows “Hamburger \* 2” in the list instead of separate entries.
3. **RecyclerView and Adapter**: The ordered items are displayed using a RecyclerView in MainActivity. The custom OrderAdapter is responsible for binding each item’s data (food name, price, quantity, and total price) to the views inside the RecyclerView, ensuring the user can view their order in a concise format.
4. **Dynamic Quantity Management**: The MainActivity checks if an item is already present in the order list. If the user adds the same item again (e.g., a second hamburger), the app increments the quantity of the existing item rather than creating a new entry, providing a more intuitive shopping experience.

**Challenges Faced**

One of the primary challenges was managing the quantity of food items efficiently. Initially, adding the same item multiple times created separate entries in the order list, which cluttered the display. To resolve this, a loop was implemented to check if the item already exists in the order list. If it does, the app increases the quantity instead of adding a new row, making the order list cleaner and more readable.

Another issue was related to the generation of the R class due to incorrect resource references. This was resolved by checking the XML layout files for errors, cleaning the project, and syncing with Gradle.

**Lessons Learned**

This project enhanced the understanding of Android components, especially RecyclerViews and Adapters. I also gained insight into how data can be dynamically added to lists in real-time and how to manage multiple activities while sharing data between them using Intent. Additionally, managing state through dynamic data structures like ArrayList provided valuable experience in handling real-world scenarios like food ordering systems.

**Future Improvements**

The current version could be improved by allowing users to adjust the quantity of items directly within the order list. Another enhancement would be to implement a checkout feature, where users can review their total order cost and confirm their purchase. Persisting data using SQLite or shared preferences would also be beneficial for saving the state when the app is closed or the device is rotated.

**Conclusion**

This project successfully implements a basic food ordering system using Android development techniques. It provides an intuitive and efficient way for users to order food, showcasing dynamic data handling and UI management through RecyclerViews and Adapters. The app can be further enhanced with additional features, making it a strong foundation for a more comprehensive mobile ordering system.