**Technologies Used:**

* React

**Technologies Approach:**

1. Built the system using React (frontend).
2. **Approach to Orders (backend):**
   1. Three array list are being used. One array is Normal Order, another array is VIP Order, final array is the combined Order.
   2. Frontend will only display the Combined Order (where it is combining VIP Order with Normal Order)
3. Most of the logic change (e.g. add normal order, add vip order etc) are built in controller. Then frontend will get the details from backend.
4. PENDING order list is set at the top level of script, acting like a centralized state. The state can then be updated via “Add Normal Order” button & “Add VIP Order” button.

**Concerns:**

1. There were two approaches to add the order into the queue:
   1. **Use a single array list:** The system needs to track which is the last VIP order, then the next VIP order will be added into the position after it in the array. **Chose this due to the easier managing of a single list**  
      **Concern:**
      1. Error-prone and hard to manage due to the complexity of the implementation.
      2. Insertion to the array is longer since the system need to shift the elements in the array to insert new VIP order
   2. **Use two array list for two Order Type and combine into one**: Easier to implement, test, check and fix if there is any error for either Order Types.   
      **Concern:** 
      1. Memory Inefficient.
      2. The runtime may be longer since the concatenation of two arrays will happen multiple times if it is being accessed frequently.
      3. When delete the order for processing, I need to delete the correct id in few order list. Otherwise, the order is not deleted properly.

**Bugs:**

1. If I add many orders and bots, then I remove one of the bot. Somehow one of the bots will process duplicate order.