**✅ 1. Update Backend (Consumer or View) to send ranked bids**

You should group bids **per requirement**, **sort them by rate ascending**, and **assign rank (L1, L2, ...)**.

Here’s how you can build this structure in your consumer:

from django.db.models import F

from asgiref.sync import sync\_to\_async

@sync\_to\_async

def get\_ranked\_bids():

from .models import Bid

bids = Bid.objects.select\_related('user', 'req').values('user\_\_username', 'req\_\_id', 'rate')

grouped = {}

for bid in bids:

req\_id = bid['req\_\_id']

if req\_id not in grouped:

grouped[req\_id] = []

grouped[req\_id].append({

'username': bid['user\_\_username'],

'rate': bid['rate']

})

# Now sort and assign ranks

result = []

for req\_id, bids\_list in grouped.items():

bids\_sorted = sorted(bids\_list, key=lambda x: x['rate']) # lower rate = better

for rank, bid in enumerate(bids\_sorted):

result.append({

'req\_id': req\_id,

'username': bid['username'],

'rate': bid['rate'],

'rank': f"L{rank+1}"

})

return result

Then in your WebSocket consumer, send this as a new message type (e.g., ranked\_bids) during connection or when a bid is submitted:

ranked\_data = await get\_ranked\_bids()

await self.send(text\_data=json.dumps({

'type': 'ranked\_bids',

'data': ranked\_data

}))

**✅ 2. Update index.html to handle ranked\_bids and show rank**

Add this inside your socket.onmessage block:

else if (data.type === 'ranked\_bids') {

const rankedBids = data.data;

rankedBids.forEach(bid => {

const targetField = document.getElementById(`rank\_info\_${bid.req\_id}\_${bid.username}`);

if (targetField) {

targetField.textContent = `Status: ${bid.rank}`;

}

});

}

Then modify your **bid section HTML** like this:

<p id="previous\_bids\_${req.id}" style="color: yellow;"></p>

Add **under it**:

<p id="rank\_info\_${req.id}\_${username}" style="color: cyan;"></p>

Where username is the **currently logged-in user’s name**, which you can pass via Django template:

const currentUser = "{{ user.username }}";

And change the above id to:

<p id="rank\_info\_${req.id}\_${currentUser}" style="color: cyan;"></p>

**✅ Example Output on the UI:**

Req ID: 10

Previous Bids: 4000 >> 5000 >> 6000

Status: L2 (means current user placed second-lowest bid)

**✅ Summary**

| **Part** | **Action** |
| --- | --- |
| Backend | Send ranked bids with usernames |
| Frontend | Match user's bid and show their rank |
| Display | <p id="rank\_info\_${req.id}\_${username}">Status: L1</p> |

Would you like me to provide the updated full index.html and consumer method for this?  
  
  
  
This also works  
bids = Bid.objects.select\_related('user', 'req') # Required here

for bid in bids:

print(bid.user.username) # No extra DB query

print(bid.req.id) # No extra DB query