1. *Column sorting and pagination :*

We are gonna implement column sorting using javascript :

First, we're gonna add some necessary bootstrap libraries needed for this and the pagination process.

```
    <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.c
ss" rel="stylesheet">

    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.
min.css" rel="stylesheet">

    <link
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.8.1/font/bootstrap-icons
.css" rel="stylesheet">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link
href="https://fonts.googleapis.com/css2?family=Mulish:wght@400;700&disp
lay=swap" rel="stylesheet">

……

……

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.
min.js"></script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.mi
n.js"></script>
```

Make the table id in **home.html** as

```
<table id="coins-table" class="table table-striped table-hover border
border-2

text-center align-middle">
```

Now update the table structure as like this :

```
……..

…….

                <tr>

                    <th onclick="sortTable(0)" style="cursor: pointer;">Name
<span id="sort-icon-name" class="bi bi-sort-up"></span></th>

                    <th onclick="sortTable(1)" style="cursor: pointer;">Image
<span id="sort-icon-image" class="bi bi-sort-up"></span></th>

                    <th onclick="sortTable(2)" style="cursor:
pointer;">Description <span id="sort-icon-description" class="bi
bi-sort-up"></span></th>

                    <th onclick="sortTable(3)" style="cursor: pointer;">Year
<span id="sort-icon-year" class="bi bi-sort-up"></span></th>

                    <th onclick="sortTable(4)" style="cursor: pointer;">Country
<span id="sort-icon-country" class="bi bi-sort-up"></span></th>

                    <th onclick="sortTable(5)" style="cursor: pointer;">Material
<span id="sort-icon-material" class="bi bi-sort-up"></span></th>

                    <th onclick="sortTable(6)" style="cursor: pointer;">Weight
<span id="sort-icon-weight" class="bi bi-sort-up"></span></th>

                    <th onclick="sortTable(7)" style="cursor: pointer;">Starting
Bid <span id="sort-icon-bid" class="bi bi-sort-up"></span></th>

                    <th onclick="sortTable(8)" style="cursor: pointer;">Coin
Status <span id="sort-icon-status" class="bi bi-sort-up"></span></th>

                    <th>Action</th>

                </tr>

……….(existing code).....
```

Now add this script along with the end of libraries in **base.html :**

```
<script>
function sortTable(columnIndex) {
  var table, rows, switching, i, x, y, shouldSwitch, dir, switchcount = 0;
  table = document.getElementById("coins-table");
  switching = true;
  dir = "asc";
  while (switching) {
    switching = false;
    rows = table.rows;
    for (i = 1; i < (rows.length - 1); i++) {
      shouldSwitch = false;
      x = rows[i].getElementsByTagName("td")[columnIndex];
      y = rows[i + 1].getElementsByTagName("td")[columnIndex];
      if (dir == "asc") {
        if (x.innerHTML.toLowerCase() > y.innerHTML.toLowerCase()) {
          shouldSwitch= true;
          break;
        }
      } else if (dir == "desc") {
        if (x.innerHTML.toLowerCase() < y.innerHTML.toLowerCase()) {
          shouldSwitch= true;
          break;
        }
      }
    }
    if (shouldSwitch) {
      rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
```

```
            switching = true;
            switchcount ++;
        } else {
            if (switchcount == 0 && dir == "asc") {
                dir = "desc";
                switching = true;
            }
        }
    }
}
</script>
```

Pagination :

To achieve this, first you need to updates in home views in **views.py** :

```
def home(request):
    if request.user.is_authenticated:
        search_params = {}
        coins_list = Coin.objects.all().order_by('-coin_id')

        if request.method == 'POST':
            # Handle search functionality and store search history
            for key in request.POST:
                if key != 'csrfmiddlewaretoken':
                    value = request.POST[key]
                    if value:
                        search_params[key] = value
                        # Save search history to the database
```

```
                SearchHistory.objects.create(user=request.user, search_text=f'{key.capitalize()}: {value}')


        # Filter coins based on search parameters
        for key, value in search_params.items():
            coins_list = coins_list.filter(**{key: value})


    paginator = Paginator(coins_list, 10)  # Change 10 to the desired number of items per page


    page = request.GET.get('page')
    try:
        coins = paginator.page(page)
    except PageNotAnInteger:
        coins = paginator.page(1)
    except EmptyPage:
        coins = paginator.page(paginator.num_pages)


    # Retrieve search history for the current user
    search_history = SearchHistory.objects.filter(user=request.user).order_by('-timestamp')


    return render(request, 'home.html', {'coins': coins, 'search_history': search_history})
  else:
    return render(request, 'home.html', {})
```

Followed by , update your **home.html** to update the front end for pagination :

(Add after table-responsive)

```
 <div class="d-flex justify-content-center mt-4">
        <ul class="pagination">
```

```
            {% if coins.has_previous %}

                <li class="page-item"><a class="page-link" href="?page={{ coins.previous_page_number }}">&laquo;</a></li>

            {% else %}

                <li class="page-item disabled"><span class="page-link">&laquo;</span></li>

            {% endif %}



            {% for i in coins.paginator.page_range %}

                {% if i == coins.number %}

                    <li class="page-item active"><span class="page-link">{{ i }}</span></li>

                {% else %}

                    <li class="page-item"><a class="page-link" href="?page={{ i }}">{{ i }}</a></li>

                {% endif %}

            {% endfor %}



            {% if coins.has_next %}

                <li class="page-item"><a class="page-link" href="?page={{ coins.next_page_number }}">&raquo;</a></li>

            {% else %}

                <li class="page-item disabled"><span class="page-link">&raquo;</span></li>

            {% endif %}

        </ul>

    </div>
```

Now you will be able to witness the table pagination in home page if there is coins and user is authenticated. As of now 10 records of coins can be viewed per page, you can change it as per your wish.

==========