1. Search listing the coins table

For creating a search implementation, we are gonna create search history to store into the database via models :

So, update your **models.py** by adding below :

```python
class SearchHistory(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    search_text = models.CharField(max_length=255)
    timestamp = models.DateTimeField(auto_now_add=True)
```

Then migrate it via terminal ,

```
python manage.py makemigrations
```

```
python manage.py migrate
```

Next, update your **views.py** with following :

```python
....
from django.db.models import F
...
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:
```

```python
                    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})

        # 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', {})
```

Now, for adding the implementation of removing the searched keyword, we need to configure the **urls.py** in coins app:

```python
from django.urls import path, include

from django.contrib.auth import views as auth_views

from coins.views import authView, home, custom_password_change,
custom_password_change_done

from django.urls import reverse_lazy

from .views import view_profile, edit_profile, create_coin,
clear_search_history



app_name = 'coins'



urlpatterns = [

  path("", home, name="home"),

  path("signup/", authView, name="authView"),
```

```python
    path("accounts/", include("django.contrib.auth.urls")),

    path("password_change/", custom_password_change,
name="password_change"),

    path("password_change/done/", custom_password_change_done,
name="password_change_done"),

    path("profile/", view_profile, name="view_profile"),

    path("profile/edit/", edit_profile, name="edit_profile"),

    path("create-coin/", create_coin, name="create_coin"),

    path("clear-search-history/<int:search_history_id>/", clear_search_history,
name="clear_search_history"),

]
```

Update the **views.py** for the definition for the function for clearing history :

```python
@login_required

def clear_search_history(request, search_history_id):

    # Retrieve the search history item to delete

    search_history_item = get_object_or_404(SearchHistory,
pk=search_history_id)


    # Check if the search history item belongs to the current user

    if search_history_item.user == request.user:

        # Delete the search history item

        search_history_item.delete()

        messages.success(request, 'Search history item deleted successfully!')

    else:

        messages.error(request, 'You do not have permission to delete this search
history item.')


    # Redirect back to the home page or any other desired page

    return redirect('coins:home')
```

Update your **home.html** with this implementation : (Add before the table)

```html
<span> </span><span> </span>

    {% if user.is_authenticated%}

    {% if search_history%}

    <div class="alert alert-info" role="alert">

        <strong>Search History:</strong>

        {% for item in search_history %}

            <span class="badge bg-warning text-dark">{{ item.search_text }} <a href="{% url 'coins:clear_search_history' search_history_id=item.id %}" class="btn btn-danger btn-sm">X</a></span>

        {% endfor %}

    </div>

    {% endif %}

    <form method="POST" action="{% url 'coins:home' %}" class="mt-3 mb-3">

        {% csrf_token %}

        <div class="row">

            <div class="col-md-3 mb-3">

                <input type="text" name="coin_name" class="form-control" placeholder="Coin Name">

            </div>

            <div class="col-md-3 mb-3">

                <input type="text" name="coin_desc" class="form-control" placeholder="Coin Description">

            </div>

            <div class="col-md-3 mb-3">

                <input type="number" name="coin_year" class="form-control" placeholder="Coin Year">

            </div>

            <div class="col-md-2 mb-3">

                <input type="text" name="coin_country" class="form-control" placeholder="Coin Country">
```

```
            </div>

            <div class="col-md-3 mb-3">
                <input type="text" name="coin_material" class="form-control" placeholder="Coin Material">
            </div>

            <div class="col-md-3 mb-3">
                <input type="number" name="coin_weight" class="form-control" placeholder="Coin Weight">
            </div>

            <div class="col-md-2 mb-3">
                <input type="number" name="starting_bid" class="form-control" placeholder="Starting Bid">
            </div>

            <div class="col-md-2 mb-3">
                <select name="coin_status" class="form-control">
                    <option value="">-- Select Coin Status --</option>
                    <option value="available">Available</option>
                    <option value="sold">Sold</option>
                    <option value="pending">Pending</option>
                </select>
            </div>

            <div class="col-md-2">
                <button type="submit" class="btn btn-primary">Search</button> <a href="{% url "coins:home" %}" class="btn btn-secondary">Reset</a>
            </div>

        </div>

    </form>

    {% endif %}
```

==