

CHECKOUT PROCESS & INVOICE

Last session, we did till adding items to the cart and saving the changes. Now next we need to build the model for the updation of order receiving and saving order item details.

Update the models.py

```
class Order(models.Model):
    # Define choices for order status
    ORDER_STATUS_CHOICES = (
        ('Pending', 'Pending'),
        ('Processing', 'Processing'),
        ('Shipped', 'Shipped'),
        ('Delivered', 'Delivered'),
        ('Cancelled', 'Cancelled'),
    )

    user = models.ForeignKey(User, on_delete=models.CASCADE)
    order_date = models.DateTimeField(auto_now_add=True)
    total_amount = models.DecimalField(max_digits=10, decimal_places=2)
    status = models.CharField(max_length=20, choices=ORDER_STATUS_CHOICES,
                              default='Pending')

class OrderItem(models.Model):
    order = models.ForeignKey(Order, on_delete=models.CASCADE)
    coin = models.ForeignKey(Coin, on_delete=models.CASCADE)
    quantity = models.PositiveIntegerField()
    rate = models.DecimalField(max_digits=10, decimal_places=2) # Add rate field
    price = models.DecimalField(max_digits=10, decimal_places=2)

    def save(self, *args, **kwargs):
        # Set the price to the rate of the associated coin multiplied by the quantity
        self.price = self.rate * self.quantity # Use rate instead of coin.rate
        super().save(*args, **kwargs)

class ShippingAddress(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    order = models.ForeignKey(Order, on_delete=models.CASCADE)
    address = models.CharField(max_length=255)
    city = models.CharField(max_length=100)
    state = models.CharField(max_length=100)
    postal_code = models.CharField(max_length=20)
```

Now, as usual migrate the models via terminal :

```
python manage.py makemigrations
```

```
python manage.py migrate
```

Now, in **cart.html**, update this along with the save changes button like this :

```
<div class="text-center">
    <button type="submit" class="btn btn-primary btn-sm">Save Changes</button>
    <button type="button" class="btn btn-success btn-sm"
onclick="checkout()">Checkout</button>
</div>
<script>
    function checkout() {
        document.getElementById('cartForm').setAttribute('method', 'post');
        document.getElementById('cartForm').setAttribute('action', "{% url 'coins:checkout'
%}");
        document.getElementById('cartForm').submit();
    }
</script>
```

Now this is for setting the url for checkout view.

Next update the app's urls.py :

```
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, edit_coin,
checkout_view
```

```
app_name = 'coins'
```

```
urlpatterns = [
    path("", home, name="home"),
    path("signup/", authView, name="authView"),
    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"),
    path("checkout/", checkout_view, name="checkout"),
```

```
]
```

Next update the views.py as follows :

```
from django.shortcuts import render, redirect, get_object_or_404 # Import render function
from Django
from django.urls import reverse
from rest_framework import status # Import status codes from Django REST Framework
from rest_framework.response import Response # Import Response class from Django REST
Framework
from rest_framework import viewsets # Import viewsets from Django REST Framework
from .models import Coin, Profile, SearchHistory, CartItem, Order, OrderItem,
ShippingAddress # Import the Coin model
from .serializers import CoinSerializer # Import the CoinSerializer
from rest_framework.views import APIView
import re
from django.contrib.auth.forms import AuthenticationForm, UserCreationForm,
PasswordChangeForm
from django.contrib.auth.decorators import login_required
from django.contrib.auth import update_session_auth_hash
from django.contrib.auth.views import PasswordChangeView, PasswordChangeDoneView
from django.urls import reverse_lazy
from django.contrib.auth.mixins import LoginRequiredMixin
from django.contrib import messages
from django.urls import reverse
from .forms import ProfileForm, CoinForm
from django.db.models import F
from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger

@login_required
def checkout_view(request):
    if request.method == 'POST':
        user = request.user
        cart_items = CartItem.objects.filter(user=user)

        # Calculate total price
        total_price = sum(item.quantity * item.coin.rate for item in cart_items)

        for item in cart_items:
            item.price = item.quantity * item.coin.rate

        if cart_items:
            return render(request, 'order_confirmation.html', {'cart_items': cart_items})
        else:
            # Redirect to the home page if the cart is empty
            return redirect(reverse('coins:home'))
    else:
        # Handle GET request (if needed)
        return redirect(reverse('coins:home'))
```

Next, you can set your template for the best view for the checkout page, in that page, we are going to make the users review and confirm their cart details and in the same page itself, place the form for collecting the shipping details from the customers for a valid checkout and order placing.

Place the `order_confirmation.html` in the `templates/` directory :

```
{% extends "base.html" %}
{% load bootstrap4 %}
{% block content %}
<div class="container mt-5">
  <div class="row">
    <div class="col-md-6">
      <h1 class="mb-4 font-weight-bold text-center">Review Your Order</h1>
      <div class="card">
        <div class="card-body">
          <h5 class="card-title">Your Order Details</h5>
          <div class="table-responsive">
            <table class="table">
              <thead>
                <tr>
                  <th scope="col">#</th>
                  <th scope="col">Coin Name</th>
                  <th scope="col">Quantity</th>
                  <th scope="col">Rate</th>
                  <th scope="col">Price</th>
                </tr>
              </thead>
              <tbody>
                {% for item in cart_items %}
                <tr>
                  <th scope="row">{{ forloop.counter }}</th>
                  <td>{{ item.coin.coin_name }}</td>
                  <td>{{ item.quantity }}</td>
                  <td>{{ item.coin.rate }}</td>
                  <td class="price">{{ item.price }}</td>
                </tr>
                {% endfor %}
              </tbody>
            </table>
          </div>
        </div>
      </div>
      <span>&nbsp;  </span>
      <h2 class="font-weight-bold total-price text-center">Total Amount : {{ total_price }}</h2>
    </div>
    <div class="col-md-6">
      <h1 class="mb-4 font-weight-bold text-center">Shipping Details</h1>
    </div>
  </div>
</div>
```

```

<div class="card">
  <div class="card-body">
    <h5 class="card-title">Shipping Address</h5>
    <form method="post" action="{% url 'coins:place_order' %}">
      {% csrf_token %}
      <div class="form-group">
        <label for="address">Address</label>
        <input type="text" class="form-control" id="address" name="address"
required>
      </div>
      <div class="form-group">
        <label for="city">City</label>
        <input type="text" class="form-control" id="city" name="city" required>
      </div>
      <div class="form-group">
        <label for="state">State</label>
        <input type="text" class="form-control" id="state" name="state"
required>
      </div>
      <div class="form-group">
        <label for="postal_code">Postal Code</label>
        <input type="text" class="form-control" id="postal_code"
name="postal_code" required>
      </div>

```

```

    </div>
  </div>
</div>
<span>&nbsp;</span>
<div class="text-center">
  <a href="{% url 'cart' %}" class="btn btn-warning btn-lg">Return</a>
  <a href="{% url 'coins:home' %}" class="btn btn-primary btn-lg">Continue
Shopping</a>
  <button type="submit" class="btn btn-success btn-lg">Place Order</button>
</div>
</div>
</form>
</div>
<script>
  document.addEventListener("DOMContentLoaded", function () {
    var prices = document.querySelectorAll('.price');
    var totalPrice = 0;

```

```

    prices.forEach(function (priceElement) {
      var price = parseFloat(priceElement.textContent.replace(/^[^0-9.-]+/g, ""));
      totalPrice += price;
    });
    document.querySelector('.total-price').textContent = "Total Amount : " +
totalPrice.toFixed(1);

```

```

    });
</script>
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, edit_coin,
checkout_view, place_order,

app_name = 'coins'

urlpatterns = [

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

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

    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"),

    path('checkout/', checkout_view, name='checkout'),

    path('place_order/', place_order, name='place_order'),

]

```

Now the confirmation page is set for the users, next we are going to implement the process for placing the order.

As the url for “Place Order” button corresponds to ‘place_order’ name, we need to configure the **urls.py** in the app’s folder :

```
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, edit_coin,
checkout_view, place_order,

app_name = 'coins'

urlpatterns = [
    path("", home, name="home"),
    path("signup/", authView, name="authView"),
    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"),
    path("checkout/", checkout_view, name='checkout'),
    path("place_order/", place_order, name='place_order'),

]
```

Now update the views for saving the order details of each and the order details.

Here is the updation code for **views.py** :

```
from .models import Coin, Profile, SearchHistory, CartItem, Order, OrderItem,
ShippingAddress

@login_required
def place_order(request):
    if request.method == 'POST':
        user = request.user
        cart_items = CartItem.objects.filter(user=user)

        # Calculate total price
```

```

total_price = sum(item.quantity * item.coin.rate for item in cart_items)

# Create an order
order = Order.objects.create(user=user, total_amount=total_price)

# Create order items
for cart_item in cart_items:
    order_item = OrderItem.objects.create(
        order=order,
        coin=cart_item.coin,
        quantity=cart_item.quantity,
        rate=cart_item.coin.rate,
        price=cart_item.price
    )

# Save shipping address
address = request.POST.get('address')
city = request.POST.get('city')
state = request.POST.get('state')
postal_code = request.POST.get('postal_code')
ShippingAddress.objects.create(user=user, order=order, address=address, city=city,
state=state, postal_code=postal_code)

# Clear the cart
cart_items.delete()

request.session['order_placed'] = True

# Redirect to the thank you page
return redirect('coins:thank_you')
else:
    # Handle GET request (if needed)
    return redirect(reverse('coins:home'))

```

Now the order received, order details for each item, shipping address were saved to the database and order is placed. Now we need to set the page for users to see the greeting and invoice of the order.

Set the url for thanks giving page , so update the app's **urls.py**

```

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, edit_coin,
checkout_view, place_order, thank_you

```



```
app_name = 'coins'
```

```
urlpatterns = [  
    path("", home, name="home"),  
    path("signup/", authView, name="authView"),  
    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"),  
    path("checkout/", checkout_view, name='checkout'),  
    path("place_order/", place_order, name='place_order'),  
    path("thank-you/", thank_you, name='thank_you'),  
]
```

Now set the **views.py** to decide what to show in thanks page :

```
from .models import Coin, Profile, SearchHistory, CartItem, Order, OrderItem,  
ShippingAddress
```

```
@login_required
```

```
def thank_you(request):
```

```
    # Check if the user just placed an order (using session or query parameter)
```

```
    if request.session.get('order_placed') or request.GET.get('order_placed'):
```

```
        # Retrieve the most recent order for the current user
```

```
        order = Order.objects.filter(user=request.user).order_by('-order_date').first()
```

```
        # Retrieve the shipping address associated with the order
```

```
        shipping_address =  
ShippingAddress.objects.filter(user=request.user).order_by('-id').first()
```

```
        request.session.pop('order_placed', None)
```

```
        return render(request, 'thank_you.html', {'order': order, 'shipping_address':  
shipping_address})
```

```
    else:
```

```
        # Redirect the user to the home page or any other appropriate page
```

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

Now you can view the invoice of the order and greetings message that the user has successfully placed the order.

Next, set the template of the page which you should place the file “**thank_you.html**” in the templates/ directory :

```
<!-- thank_you.html -->
{% extends "base.html" %}
{% load bootstrap4 %}
{% block content %}
<div class="container mt-5">
  <h1 class="mb-4 font-weight-bold text-center">Thank You for Your Order!</h1>
  <p class="lead text-center">Your order has been successfully placed.</p>
  <div class="card">
    <div class="card-body">
      <h5 class="card-title">Order Invoice</h5>
      <p><strong>Order ID:</strong> {{ order.id }}</p>
      <p><strong>Order Date:</strong> {{ order.order_date }}</p>
      <p class="text-danger font-weight-bold"><strong>Total Amount:</strong> {{
order.total_amount }}</p>
      <p><strong>Shipment Status :</strong> {{ order.status }}</p>
      <hr>
      <h5 class="card-title">Shipping Address</h5>
      <p><strong>Address:</strong> {{ shipping_address.address }}</p>
      <p><strong>City:</strong> {{ shipping_address.city }}</p>
      <p><strong>State:</strong> {{ shipping_address.state }}</p>
      <p><strong>Postal Code:</strong> {{ shipping_address.postal_code }}</p>
      <hr>
      <h5 class="card-title">Ordered Items</h5>
      <table class="table">
        <thead>
          <tr>
            <th scope="col">Item</th>
            <th scope="col">Quantity</th>
            <th scope="col">Price</th>
          </tr>
        </thead>
        <tbody>
          {% for item in order.orderitem_set.all %}
          <tr>
            <td>{{ item.coin.coin_name }}</td>
```

```

        <td>{{ item.quantity }}</td>
        <td>${{ item.price }}</td>
    </tr>
{% endfor %}
</tbody>
</table>
</div>
</div>
<span>&nbsp;</span>
<div class="text-center">
    <a href="{% url 'coins:home' %}" class="btn btn-secondary">Go To Home</a>
    <a href="{% url 'coins:my_orders' %}" class="btn btn-warning">My Orders</a>
</div>
</div>
{% endblock %}

```

Now , the last end of the order page has been reached and finished , now we need to set the link for the “My Orders” page.

Set the app’s **urls.py** :

```

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, edit_coin,
checkout_view, place_order, thank_you, my_orders

app_name = 'coins'

urlpatterns = [
    path("", home, name="home"),
    path("signup/", authView, name="authView"),
    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"),
    path('checkout/', checkout_view, name='checkout'),
    path('place_order/', place_order, name='place_order'),
    path('thank-you/', thank_you, name='thank_you'),
    path('my-orders/', my_orders, name='my_orders'),

```

]

Update the **views.py** to configure what to show in “My Orders” Page :

```

from .models import Coin, Profile, SearchHistory, CartItem, Order, OrderItem,
ShippingAddress

```

```

@login_required
def my_orders(request):
    # Retrieve orders associated with the current user
    orders = Order.objects.filter(user=request.user).order_by('-order_date')
    return render(request, 'my_orders.html', {'orders': orders})

```

Set the template for the “My Orders” Page, place the **my_orders.html** in templates/ directory :

```

{% extends "base.html" %}
{% load bootstrap4 %}

{% block content %}
<div class="container mt-5">
    <h1 class="mb-4 font-weight-bold text-center">My Orders <a href="{% url 'coins:home' %}" class="btn btn-warning btn-sm">Return</a></h1>
    {% if orders %}
    <div class="row">
        {% for order in orders %}

```

```

<div class="col-md-4 mb-4">
    <div class="card h-100">
        <div class="card-body">
            <h5 class="card-title">Order #{{ order.id }}</h5>
            <p class="card-text">Order Date: {{ order.order_date }}</p>
            <p class="card-text">Total Amount: ${{ order.total_amount }}</p>
            <p class="card-text">Shipment Status: {{ order.status }}</p>
            <a href="{% url 'coins:order_details' order.id %}" class="btn
btn-primary">View Details</a>
        </div>
    </div>
</div>
{% endfor %}
</div>
{% else %}
    <p class="lead text-center">You have no orders yet.</p>
{% endif %}
</div>
{% endblock %}

```

You can place the url link for “My Orders” page in base.html by placing {% url ‘coins:my_orders’ %} in the appropriate place in navbars.

As the next step, set the url for viewing details for each order items in “My Orders” page.

Now, set the app’s **urls.py** as :

```

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, edit_coin,
checkout_view, place_order, thank_you, my_orders, order_details

app_name = 'coins'

```

```
urlpatterns = [
    path("", home, name="home"),
    path("signup/", authView, name="authView"),
    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"),
    path('checkout/', checkout_view, name='checkout'),
    path('place_order/', place_order, name='place_order'),
    path('thank-you/', thank_you, name='thank you'),
    path('my-orders/', my_orders, name='my orders'),
    path('orders/<int:order_id>/', order_details, name='order_details'),

]
```

Update the **views.py** to display the invoice details in the details page.

```
from .models import Coin, Profile, SearchHistory, CartItem, Order, OrderItem,
ShippingAddress

@login_required
def order_details(request, order_id):
    # Retrieve the order object from the database
    order = get_object_or_404(Order, id=order_id)

    # Retrieve the associated shipping address for the order
    shipping_address = ShippingAddress.objects.filter(order=order).first()

    # Render the order details page with order and shipping address information
    return render(request, 'order_details.html', {'order': order, 'shipping_address':
shipping_address})
```

Now set the template design to view the order details by placing the file “**order_details.html**” in the templates/ directory :

```
{% extends "base.html" %}
{% load bootstrap4 %}
{% block content %}
    <div class="container mt-5">
        <h1 class="text-center mb-4"><strong>COIN DETAILS</strong></h1>
        <div class="row justify-content-center">
            <div class="col-md-12">
                {% bootstrap_messages %}
                <div class="card border-0 shadow-lg" style="background-color: #c7e619;">
                    <div class="card-body">
                        <h3 class="card-title text-center mb-4">{{ coin.coin_name }}</h3>
                        <div class="row">
                            <div class="col-md-6">
                                <p class="card-text"><strong>Year:</strong> {{ coin.coin_year }}</p>
                                <p class="card-text"><strong>Country:</strong> {{ coin.coin_country }}</p>
                                <p class="card-text"><strong>Material:</strong> {{ coin.coin_material }}</p>
                                <p class="card-text"><strong>Weight:</strong> {{ coin.coin_weight }}</p>
                                <p class="card-text"><strong>Starting Bid:</strong> {{ coin.starting_bid }}</p>
                                <p class="card-text"><strong>Coin Status:</strong> {{ coin.coin_status }}</p>
                                <span>&nbsp;</span>
                                <h3 class="text-danger font-weight-bold"><strong>Coin Rate</strong> {{ coin.rate }}</h3>
                            </div>
                            <div class="col-md-6">
                                {% if coin.coin_image %}
                                    <p class="text-center"></p>
                                {% else %}
                                    <p class="text-center">No Image</p>
                                {% endif %}
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
```

```

{% endif %}
<p class="card-text"><strong>Description:</strong> {{ coin.coin_desc
}}</p>
</div>
</div>
</div>
</div>
<div>&nbsp;</div>
<div class="text-center">
<div class="btn-group">
<form action="{% url 'coins:home' %}">
<button type="submit" class="btn btn-secondary">Back to
Home</button>
</form><span>&nbsp;</span>
{% if request.user.id == coin.created_by_id %}
<form action="{% url 'edit_coin' coin_id=coin.coin_id %}">
<button type="submit" class="btn btn-warning">Edit Coin</button>
</form><span>&nbsp;</span>
<form method="post" action="{% url 'add_to_cart' coin.coin_id %}">
{% csrf_token %}
<button type="submit" class="btn btn-success">Buy Now</button>
</form>
{% endif %}
</div>
</div>
</div>
</div>
</div>
{% endblock content %}

```

Now , you are set with the checkout process with all ends to view the necessary pages and make necessary restrictions as flexible to the users.

