Django application

app_gen

views.py

```
from django.shortcuts import render
from app_gen.models import Product
from django.core.paginator import Paginator

# Create your views here.
def index(request):
    products = Product.objects.filter(trending=True)
    return render(request, 'index.html', {'products': products})

def about(request):
    return render(request, 'about.html')

def contact(request):
    return render(request, 'contact.html')

def productDetail(request, id):
    product = Product.objects.get(pk=id)
    return render(request, 'detail.html', {'product': product})

def products(request):
    all_products = Product.objects.all().order_by('name')
    page = request.GET.get('page')
    paginator = Paginator(all_products,9) # number of products
    all_products = paginator.get_page(page)
    return render(request, 'products.html', {'all_products': all_products})
```

สร้างเทมเพลทหน้า Home, Product, Product detail, About, Contact และแสดง paginator คือการแบ่งหน้า หรือเลขหน้า

urls.py

```
from django.urls import path
from app_gen import views

urlpatterns = [
    path('', views.index),
    path('product/details/<int:id>', views.productDetail, name='productDetail'),
    path('products', views.products),
    path('about', views.about),
    path('contact', views.contact),
]
```

กำหนด URL patterns ที่เชื่อมโยงระหว่าง URL และ views

models.py

```python
from django.db import models

# Create your models here.
class Product(models.Model):
    name = models.CharField(max_length=50)
    description = models.TextField()
    price = models.DecimalField(max_digits=10,decimal_places=2)
    stock = models.IntegerField()
    trending = models.BooleanField(default=False)
    image = models.ImageField(upload_to='products', blank=True)
```

สร้างโมเดล Product ประกอบไปด้วย name, description, price, stock, trending, image

admin.py

```python
from django.contrib import admin
from app_gen.models import Product

class ManageProduct(admin.ModelAdmin):
    list_display = ['name', 'price','stock','trending']
    list_editable = ['price', 'stock', 'trending']
    list_per_page = 6
    search_fields = ['name']
    list_filter = ['trending']

# Register your models here.
admin.site.register(Product, ManageProduct)
```

นำข้อมูลจากโมเดลไปจัดการที่หน้า admin

app_order

views.py

```python
from django.shortcuts import render
from django.contrib.auth.decorators import login_required
from app_cart.models import Cart, CartItem
from app_order.models import Order, OrderDetail
from app_gen.models import Product
from app_cart.views import create_cartId


# Create your views here.
login_required(login_url = '/login')
def order(request):
    if request.method == 'POST':
        fullname = request.POST['fullname']
        phone = request.POST['phone']
        address = request.POST['address']
```

```
        cart = Cart.objects.get(cart_id=create_cartId(request),
customer=request.user)
        cartItem = CartItem.objects.filter(cart=cart)
        total = 0
        for item in cartItem:
            total += (item.product.price * item.quantity)
        #create Purchase order
        order = Order.objects.create(
            fullname = fullname,
            phone = phone,
            address = address,
            total = total,
            customer = request.user
        )
        order.save()
        for item in cartItem:
            order_detail = OrderDetail.objects.create(
                product = item.product.name,
                quantity = item.quantity,
                price = item.product.price,
                order = order
            )
            order_detail.save()
            #Stock cut
            product = Product.objects.get(pk=item.product.id)
            product.stock = int(item.product.stock-order_detail.quantity)
            product.save()
            item.delete()
        cart.delete()
        return render(request, 'order_success.html')
    else:
        return render(request, 'order.html')

login_required(login_url = '/login')
def history(request):
    orders = Order.objects.filter(customer = request.user)
    return render(request, 'history.html', {'orders': orders})

login_required(login_url = '/login')
def orderDetail(request, order_id):
    order = Order.objects.get(pk=order_id)
    if order.customer == request.user:
        order_items = OrderDetail.objects.filter(order=order)
        return render(request, 'order_detail.html', {'order':order,
'order_items':order_items} )
    return render(request, 'order_detail.html')
```

order: สร้างรายการสั่งซื้อรับค่า fullname phone address และข้อมูลในตะกร้าสินค้า คือ product quantity price และ order บันทึกและตัดสต็อกเมื่อทำการสั่งซื้อ โดยต้องมีการ login

history: รับรายการ order เพื่อนำไปแสดง โดยต้องมีการ login

orderDetail: นำรายละเอียด order มาแสดงเมื่อมีการร้องขอจากผู้ใช้งาน โดยต้องมีการ login

urls.py

```
from django.urls import path
from app_order import views

urlpatterns = [
    path('order', views.order),
    path('history', views.history),
    path('order/<int:order_id>' , views.orderDetail, name='orderDetail'),
]
```

กำหนด URL patterns ที่เชื่อมโยงระหว่าง URL และ views

models.py

```
from django.db import models
from django.contrib.auth.models import User

# Create your models here.
class Order(models.Model):
    fullname = models.CharField(max_length=255, blank=True)
    phone = models.CharField(max_length=30, blank=True)
    address  = models.CharField(max_length=255, blank=True)
    total = models.DecimalField(max_digits=10, decimal_places=2)
    created = models.DateTimeField(auto_now_add=True)
    customer = models.ForeignKey(User, on_delete=models.CASCADE, default=None)

class OrderDetail (models.Model):
    product = models.CharField(max_length=255)
    price = models.DecimalField(max_digits=10, decimal_places=2)
    quantity = models.IntegerField()
    created = models.DateTimeField(auto_now_add=True)
    order = models.ForeignKey(Order, on_delete=models.CASCADE)

    def sub_total(self):
        return self.price * self.quantity
```

สร้างโมเดล Order ประกอบไปด้วย fullname, phone, address, total, created, customer โมเดล OrderDetail ประกอบไปด้วย product, price, quantity, created, order และ ค่าราคาและจำนวนสินค้าของแต่ละรายการ

app_cart

views.py

```
from django.shortcuts import render, redirect
from django.contrib.auth.decorators import login_required
from app_gen.models import Product
from app_cart.models import Cart, CartItem

# Create your views here.
def create_cartId(request):
    cart = request.session.session_key
    if not cart:
        cart = request.session.create()
    return cart

@login_required(login_url='/login')
def removeCart(request, product_id):
    cart = Cart.objects.get(cart_id=create_cartId(request), customer =
request.user)
    product = Product.objects.get(pk=product_id)
    cartItem = CartItem.objects.get(product = product, cart = cart) #target
    cartItem.delete()
    return redirect('/cart')

@login_required(login_url='/login')
def cart(request):
    counter = 0
    total = 0
    try:
        cart = Cart.objects.get(cart_id = create_cartId(request), customer =
request.user)
        cartItem = CartItem.objects.filter(cart = cart)
        for item in cartItem:
            counter += item.quantity
            total += (item.product.price * item.quantity)
    except (Cart.DoesNotExist, CartItem.DoesNotExist):
        cart = None
        cartItem = None
    return render(request, 'cart.html', {'cartItem':cartItem, 'total':total,
'counter':counter})

@login_required(login_url='/login')
def addCart(request, product_id):
    product = Product.objects.get(pk=product_id)

    try:
        cart = Cart.objects.get(cart_id=create_cartId(request),
customer=request.user)
    except Cart.DoesNotExist:
        cart = Cart.objects.create(
            cart_id=create_cartId(request),
            customer=request.user
        )
        cart.save()
    try:
        cartitem = CartItem.objects.get(product=product, cart=cart)
```

```
        if cartitem.quantity < product.stock:  # Check if quantity can be
increased
            cartitem.quantity += 1
            cartitem.save()
    except CartItem.DoesNotExist:
        if product.stock > 0:  # Check if there's available stock
            cartitem = CartItem.objects.create(
                product=product,
                cart=cart,
                quantity=1
            )
            cartitem.save()

    return redirect('/cart')
```

create_cartId: สร้างคีย์สำหรับตะกร้าสินค้า หรือ cartId

removeCart: ลบสินค้าในตะกร้า ตาม product ที่เลือก โดยต้องมีการ login

cart: นับจำนวน และแสดงราคารวมของสินค้าแต่ละชิ้น โดยต้องมีการ login

addCart: สร้างตะกร้าสินค้า และตรวจสอบจำนวนสินค้าที่เพิ่มกับสินค้าที่เหลือในคลัง โดยต้องมีการ login

urls.py

```
from django.urls import path
from app_cart import views

urlpatterns = [
    path('cart', views.cart),
    path('cart/add/<int:product_id>', views.addCart,name='addCart'),
    path('cart/remove/<int:product_id>', views.removeCart,name='removeCart')
]
```

กำหนด URL patterns ที่เชื่อมโยงระหว่าง URL และ views

admin.py

```
from django.contrib import admin
from app_cart.models import Cart

# Register your models here.
admin.site.register(Cart)
```

นำค่าจาก Cart ไปจัดการที่หน้า admin

models.py

```
from django.db import models
from django.contrib.auth.models import User
from app_gen.models import Product


# Create your models here.
class Cart(models.Model):
    cart_id = models.CharField(max_length=255, blank=True)
    customer = models.ForeignKey(User, on_delete=models.CASCADE, default=None)

class CartItem(models.Model):
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    cart = models.ForeignKey(Cart, on_delete=models.CASCADE)
    quantity = models.PositiveIntegerField()

    def sub_total(self):
        return self.product.price * self.quantity
```

สร้างโมเดล Cart ประกอบไปด้วย cart_id และ customer โมเดล CartItem ประกอบไปด้วย product cart และ quantity และราคาสินค้ารวม

app_user

views.py

```
from django.shortcuts import render, redirect
from django.contrib import messages
from django.contrib.auth.models import User, auth

def register(request):
    if request.method == 'POST':
        username = request.POST['username']
        email = request.POST['email']
        password = request.POST['password']
        if  username=='' or email=='' or password=='':
            messages.warning(request,'Please enter all fields')
            return redirect('/register')
        else:
            if User.objects.filter(username=username).exists():
                messages.warning(request,'The user already exists')
                return redirect('/register')
            else:
                user=User.objects.create_user(
                    username=username,
                    email=email,
                    password=password,
                )
                user.save()
                messages.success(request, 'Create account successfully')
                return redirect('/register')
    else:
```

```
        return render(request, 'register.html')

def login(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        if username=='' or password=='':
            messages.warning(request, 'Please enter all fields.')
            return redirect('/login')
        else:
            user=auth.authenticate(username=username, password=password)
            if user is not None:
                auth.login(request, user)
                return redirect('/')
            else:
                messages.warning(request,'Account information not found.')
                return redirect('/login')
    else:
        return render(request, 'login.html')

def logout(request):
    auth.logout(request)
    return redirect('/login')
```

register: เก็บค่า username email และ password โดยต้องไม่เว้นว่าง และ username ไม่ซ้ำกับที่มีในระบบ

login: เก็บค่า username และ password โดยต้องไม่เว้นว่าง เช็คความถูกต้องกับค่าที่บันทึกในระบบ

login: ออกจากระบบ

urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('register/', views.register, name='register'),
    path('login/', views.login, name='login'),
    path('logout/', views.logout, name='logout'),
    path('profile/', views.profile, name='profile'),
    path('edit_profile/', views.edit_profile, name='edit_profile'),
]
```

กำหนด URL patterns ที่เชื่อมโยงระหว่าง URL และ views