# **الخطوه الاولى**

**عند بدايه كل مشروع**

# **V num #2**

# **\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

1. تحميل virtualenv على الجهاز

pip install virtualenv

1. انشاء بيئه افتراضيه

virtualenv lib

**حيث ان lib اسم البيئه الافتراضيه**

1. تفعيل البيئه الافتراضيه

*Lib/Scripts/activate*

**حيث lib اسم المبيئه الافتراضيه و ال Scripts اول حرف caps**

1. تحميل ال Django

*pip install django*

1. *انشاء المشروع*

django-admin startproject lms

**حيث lms هى اسم المشروع**

1. *تشغيل السيرفر*

**اول مره لازم تعمل الكود ده**

*python manage.py migrate*

**ثم فى كل مره تعمل ده على طول**

python manage.py runserver

# ***########################################***

# **الخطوه الثانيه**

**انشاء تطبيق**

# **V num #3**

# **\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

# *فى ال terminal نكتب*

# *python manage.py startapp lms\_app*

# *فى ملف المشروع ندخل settings.py ثم ندخل الى INSTALLED\_APPS*

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'lms\_app',

]

# ***########################################***

# **الخطوه الثالثه**

**انشاء رابط للتطبيق**

# **V num #4**

# **\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

# *فى folder الخاص بالتطبيق نعمل file و نسميه urls.py بحيث اى صفحه يتم انشائها يتم ربطها بالتطبيق من خلال هذا ال file*

1. *نذهب الى file باسم urls.py ولاكن الخاص بالمشروع الرئيسى*
2. *نقوم بعمل import لل include  
   from Django.urls import path, include*
3. *ثم نذهب الى ال list باسم urlpatterns و نضيف هذا الكود  
   path(‘’, include(‘lms\_app.urls’)),***ملحوظه تركنا ال ‘’ فارغه بالتالى سوف يتم الذهاب الى هذا التطبيق مباشره بمجرد فتح الموقع واذا اضفت بداخلها كلمات سوف يتوجب على اضافه تلك الكلمات الى لينك الموقع اذا اردت الذهاب اليها**
4. *فى file المسمى ب urls.py الخاص بالتطبيق نقوم بربط الصفح الخاصه بالتطبيق كما هو موضح فى الخطوات التاليه*

# ***########################################***

# **الخطوه الرابعه**

**انشاء templates**

**انشاء صفحه**

# **V num #5**

# **\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

**انشاء ال templates و ربطها**

# *بمحازات folders المشروع و ال التطبيق نقوم بانشاء folder و نسميه templates*

1. *بداخل folder ال templates نقوم بانشاء file من نوع html و نسميه على سبيل المثال index.html*
2. *نجعل ال Django يرى folder ال templates كالتالى*
3. *ندخل فى ال settings.py الخاص بالمشروع*
4. *نعمل import os*
5. *نبحث عن TEMPLATES*
6. *بداخل ال LIST الخاص ب TEMPLATES سوف نجد key باسم DIRS نكتب بداخل ال value الخاص بها  
   os.path.join(BASE\_DIR, ‘templates’)*

**انشاء الصفحه**

# *نذهب الى ال views.py الخاصه بالتطبيق و نقوم بانشاء الصفحه الداله function from Django.shortcuts import render def index(request): return render(request, ‘index.html’)*

# **ربط الصفحه**

*ننشا file باسم urls.py فى folder التطبيق و نكتب به*   
*from Django.urls import path  
From . import views*

*Urlpatterns = [*

*Path(‘’, views.index, name= ‘index’),*

*]*

# ***########################################***

# **الخطوه الخامسه**

**اضافه الفرونت اند add front end**

# **V num #6**

# **\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

1. *تحميل البيانات*
2. *نقوم بانشاء file باسم base.html او اى اسم مناسب و ذلك لنضع به المحتوى الاساسى لكل صفحه حتى نورث منه و نوفر وقد و كود*
3. *نفتح احد الصفحات الخاصه ب data الخاصه بال front end و يفضل صفحه index*
4. *نعمل cut من بدايه الصفحه و ناخد معانا فتحه tag الخاصه ب ال body ثم نعمل paste داخل ال base.html*
5. *نعمل cut من نهايه الصفحه وصولا لفوق script و ثم نعمل paste داخل base.html بحيث تكون فى الاسفل*
6. *نعمل فى ال templates مجلد folder و نسميه parts و بداخل ال parts نعمل file نسميه nav.html و اخر sidebar.html و اخر footer.thml***خد بالك و انت بتاخد الفوتر سيب div الى تحتيها انها تبع المحتوى نفسه**
7. *نعمل folder نسميه pages جواه file نسميه index.html ده لو انت مكنتش عملتها و تحط فيه باقى المحتوى ما عدا ال div الى فى الاخر هنسيبه و نحطه فى ال base.html لانه لازم يكون تحت ال footer*
8. *فى ال base.html نعمل بين بدايه الصفحه و نهايتها block علشان يكون فيه المحتوى و اى صفحه هيكون فيها محتوى ذى index.html نحط المحتوى داخل هاذا ال block  
   {% block content %}  
   {% endblock %}  
   ثم نحط ال div الخاصه ب قفله المحتوى تحت ال endblock  
   ثم نعمل tag الميراث من ال base.html فى الصفحه الى هتتعرض بداخله  
   {% extends ‘base.html’ %}*
9. *ما فعلناه ب index.html هنعمله ب books.html لاكن هذه المره هاخد المحتوى فقط*
10. *اضافه ال footer, nav, sidebar الى ال base.html  
    مبين الفتحه ال body و بدايه ال block الخاص بالمحتوى نحط ال nav  
    {% include ‘parts/nav.html’ %}  
    ثم ال sidebar  
    {% include ‘parts/sidebar.html %}  
    ثم بين قفله ال endblock و ال div الى بتقفل المحتوى نحط ال footer  
    {% include ‘parts/footer.html’ %}*
11. *فى ال views الخاصه بالتطبيق نعمل function لعرض كل صفحه ولا تنسى اضافه pages/ لان الصفحات ب templates محتوطه جوه folder باسم pages مثال  
    def index(request):  
        return render(request, 'pages/index.html')*
12. *ثم نربط كل صفحه ب urls.html الخاصه بالتطبيق  
    urlpatterns = [  
        path('', views.index, name='index'),  
        path('books', views.books, name='books'),  
        path('delete', views.delete, name='delete'),  
        path('update', views.update, name='update'),  
    ]*
13. *ربط ال css*
14. *ناخد copy للملفات bootstrap, dist, plugins ثم داخل folder الخاص بالمشروع الذى بداخل manage.py سوف تجد folder اخر باسم المشروع ننشئ بداخله folder باسم static و بداخله نعمل paste*
15. *نذهب الى ال sittings الخاصه بالمشروع و نبحث عن static files (css, java*
16. *نكتب   
    STATIC\_ROOT = os.path.join(BASE\_DIR, ‘static’)  
    STATIC\_URL = ‘/static/’  
    STATICFILES\_DIRS = [  
    os.path.join(BASE\_DIR, ‘lms/static’),  
    ]*
17. *ثم فى ال terminal نكتب  
    python manage.py collectstatic*
18. *ثم فى صفحه ال base.html  
    {% load static %}*
19. *اى لينك ربط او صور نحطه بين  
    “{% static ‘ the link ‘ %}”  
    وعلشان تكون سهله استعمل replace control+h  
    و راجع على الى حصل و صلح اى خطأ*

***####################################***

# **الخطوه السادسه**

**اضافه الصور الاستاتيك static img**

# **V num #10**

# **\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

1. *فى الصفحه التى بداخلها الصور static التى اريد عرضها على ثبيل المثال books.html  
   {% load static %}*
2. *اى لينك ربط او صور نحطه بين  
   “{% static ‘ the link ‘ %}”*

***####################################***

# **الخطوه السابعه**

**اضافه قاعده البيانات data base**

# **V num #12**

# **\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

1. *نذهب الى التطبيق الى هعمله قاعده بيانات داخل models.py*
2. *نقوم بانشاء class اى جدول للعناصر المراد اضافتها فى قاعده البيانات كالتالى*

from django.db import models

# Create your models here.

class Category(models.Model):

    name = models.CharField(max\_length=50)

class Book(models.Model):

    status\_book = [

        ('availble', 'availble'),

        ('rental', 'rental'),

        ('sold', 'sold'),

    ]

    title = models.CharField(max\_length=250)

    author = models.CharField(max\_length=250, null=True, blank=True)

    photo\_book = models.ImageField(upload\_to='photo', null=True, blank=True)

    photo\_author = models.ImageField(upload\_to='photo', null=True, blank=True)

    pages = models.IntegerField(null=True, blank=True)

    price = models.DecimalField(max\_digits=5, decimal\_places=2, null=True, blank=True)

    retal\_price\_day = models.DecimalField(max\_digits=5, decimal\_places=2, null=True, blank=True)

    retal\_period = models.IntegerField(null=True, blank=True)

    active = models.BooleanField(default=True)

    status = models.CharField(max\_length=50, choices=status\_book,null=True, blank=True)

    category = models.ForeignKey(Category, on\_delete=models.PROTECT, null=True, blank=True)

**فى الجزئيه الخاصه ب on delete دى تعنى عند المسح نمسح الجدول بتاع ال category ولا لا**

**و فى حاله PROTECT يبقى لا طب لو عاوز امسح يبقى CASCADE**

1. *اذا قمنا باستخدام ImageField نزهب الى TERMINAL و نكتب*

*Pip install pillow*

*ثم ندخل الجدول فى ال migrations*

*Python manage.py makemigrations*

*ثم ندخل الجدول فى قاعده البيانات*

*Python manage.py migrate*

***####################################***

# **الخطوه الثامنه**

**admin**

# **V num #13**

# **\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

1. فى ال terminal نكتب Python manage.py createsuperuser
2. ادخل الاسم
3. ادخل ايمال
4. ادخل باصورد الباصورد مش هيظهر معاك و انت بتكتب
5. كرر الباصورد
6. ادخل y لو قال ان الباصورد ضعيف
7. لازم نضيف الجداول class فى صفحه ال admin الخاصه بالتطبيق  
   فى صفحه ال admin نكتب  
   from Django.contrib import admin  
   from .models import \*  
     
   admin.site.register(Book) نلاحظ ان Book هى اسم ال class  
   admin.site.register(Category)
8. علشان الكتاب مظهرش فى ال admin باسم object نعمل function فى نهايه ال class كالتالى

class Category(models.Model):

    name = models.CharField(max\_length=50)

    def \_\_str\_\_(self):

        return self.name

class Book(models.Model):

    status\_book = [

        ('availble', 'availble'),

        ('rental', 'rental'),

        ('sold', 'sold'),

    ]

    title = models.CharField(max\_length=250)

    author = models.CharField(max\_length=250, null=True, blank=True)

    photo\_book = models.ImageField(upload\_to='photo', null=True, blank=True)

    photo\_author = models.ImageField(upload\_to='photo', null=True, blank=True)

    pages = models.IntegerField(null=True, blank=True)

    price = models.DecimalField(max\_digits=5, decimal\_places=2, null=True, blank=True)

    retal\_price\_day = models.DecimalField(max\_digits=5, decimal\_places=2, null=True, blank=True)

    retal\_period = models.IntegerField(null=True, blank=True)

    active = models.BooleanField(default=True)

    status = models.CharField(max\_length=50, choices=status\_book,null=True, blank=True)

    category = models.ForeignKey(Category, on\_delete=models.PROTECT, null=True, blank=True)

    def \_\_str\_\_(self):

        return self.title

1. لو فى صور فى ال data لازم نعمل اعداتات المديا
2. نذهب الى ال sittings.py الخاص بالمشروع و نذهب الى نفس مكان ال static files (
3. و تحت اعدادات ال static نعمل كمنت علشان نفصل ثم نكتب  
   MEDIA\_ROOT = os.path.join(BASE\_DIR, ‘media’)  
   MEDIA\_URL = ‘/media/’
4. ثم نذهب الى urls.py الخاص بالمشروع و نضيف التالى   
   from Django.conf import settings  
   from Django.conf.urls.static import static
5. ثم عند ال list الخاص ب urlpatterns نضيف + static() كالتالى   
   urlpatterns = [

Path(‘admin/’, admin.site.urls),  
path(‘’, include(‘lms\_app.urls’)),  
] + static(settings.MEDIA\_URL , document\_root = settings.MEDIA\_ROOT)

***####################################***

# **الخطوه التاسعه**

**عرض الكتب التى بقاعده البيانات فى الموقع**

# **V num #15**

# **\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

1. فى التطبيق داخل ال views.py
2. نقوم باستدعاء ال class من ال models.py  
   *from . models import \**
3. ثم نضيف بداخل ال function الخاصه بالصفحه التى سوف اعرض بها البيانات المراده التالى
4. Dictionary لها key من اختيارى و value كالتالى ال class ثم objects ثم all()  
   *Book.objects.all(),*
5. نضيف فى ال return اسم ال dictionary ليصبح كالتالى   
   *def index(request):  
   context = {  
   ‘books’: Book.objects.all(),  
   }  
   return render(request, ‘pages/index.html’, context)*
6. ثم نذهب الى الصفحه التى سوف تظهر فيها البيانات و نقوم بمسح كل الكتب الافتراضيه الموضوعه و نترك كتاب واحد فقط و نضعه بين tags الخاصه ب for loop
7. لوضع بيانات كل كتاب ندخل داخل الكتاب الذى تم وضعه فى ال for loop و نبحث عن المكان المناسب لوضع البينات مثال :  
   لو عايز احط عنوان ابحث عن مكان المناسب للعنوان و اضع فيه اثنان {{}} داخل بعض و قوم باستخدام ال object لاستخراج البيانات التى فيه مثال  
   {{book.title}}  
   **ملحوظه book ده الى انا مستعمله فى اللوب  
   فى حاله الصوره بنعمل نفس الشئ لاكن جوه ال Src = “” ولازم نحط فى الاخر .url كما هو بالمثال**<img style="width:100px; height: 100px;" class="img-circle elevation-2" src="{{book.photo\_author.url}}" alt="User Avatar">
8. صوره الكتاب هندخله يدوى ب css  
   جوه ال div المسئوله عن خلفيه الكتاب نزود   
   <div style="background: url({{book.photo\_book.url}}); background-size:cover" class="widget-user-header bg-info">
9. لو عاوز اقله انه لو مفيش صوره للمؤلف متعملش حاجه او اعمل حاجه مختلفه  
   {% if book.photo\_author %}

                  <img style="width:100px; height: 100px;" class="img-circle elevation-2" src="{{book.photo\_author.url}}" alt="User Avatar">

                {% endif %}

1. نغير الوان ال available او sold  
    خد بالك لما بتعمل if block و بتقول انه على سبيل المثال *book.status == 'availble'  
   لازم الكلمه الى بعد == تكون بين علامات تنصيص “” لانها من نوع string*
2. *نغير سعر البيع او الاجار*
3. *نضيف التصنيفات من ال data زى ما تضفت الكتب ب ال for loop  
   لو عايز فى القايمه بتعتك تظهر عدد معين من العناصر بتعمل التالى*{% for cat in category|slice:'4' %}

                <li class="nav-item ">

                  <a id="" class="nav-link bg-primary" href="#">{{cat.name}}</a>

                </li>

                {% endfor %}

***####################################***

# **الخطوه العاشره**

**عرض بيانات ال sidebar فى كل صفح الموقع**

# **V num #17**

# **\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

ببساطه هنعمل الوب بتعنا عادى و نضيف جواه اسم المتغير بتاع ال object جوه {{ }} ذى مكنت بعمل فى عرض الكتب فى صفحه ال index لاكن لا تنسى ان تعمل فى الvieds.py نسخه من dictionary الخاص بعرض ال data جوه ال function الخاص بكلا من books و delete و update

مثال :

from django.shortcuts import render

from . models import \*

# Create your views here.

def index(request):

    context = {

        'books': Book.objects.all(),

        'category': Category.objects.all(),

    }

    return render(request, 'pages/index.html', context)

def books(request):

    context = {

        'books': Book.objects.all(),

        'category': Category.objects.all(),

    }

    return render(request, 'pages/books.html', context)

def delete(request):

    context = {

        'books': Book.objects.all(),

        'category': Category.objects.all(),

    }

    return render(request, 'pages/delete.html', context)

def update(request):

    context = {

        'books': Book.objects.all(),

        'category': Category.objects.all(),

    }

    return render(request, 'pages/update.html', context)

نفس الشئ عند عرض الكتب فى صفحه ال books

***####################################***

# **الخطوه الحاديه عشر**

**اضافه بيانات من المستخدم من خلال صفحه الويب**

# **V num #19**

# **\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\**

1. جوه التطبيق بنعمل file بنسميه forms.py
2. جوه forms.py نعمل import  
   *from Django import forms*
3. نعمل import لل class الى كنا عملنها فى ال data الى باسم Book و Category  
   *from . models import Book, Category*
4. نعمل class  
   *class BookForm(forms.ModelForm):  
   class Meta:  
   model = Book  
   fields = ‘\_\_all\_\_’ او لو عايز خانات معينه و هذا افضل fields = [‘title’, ‘author’, ‘category’, …]  
   widgets = {  
   ‘title’: forms.TextInput(attrs={‘class’ : ‘form-control’}),  
   ‘author’: forms.TextInput(attrs={‘class’ : ‘form-control’}),  
   ‘category’: forms.Select(attrs={‘class’ : ‘form-control’}),  
   ‘price’: forms.NumberInput(attrs={‘class’ : ‘form-control’}),  
   ‘photo\_author’: forms.FileInput(attrs={‘class’ : ‘form-control’}),  
   }*
5. نقوم باضافه هذا ال class فى ال functions الموجوده فى ال views  
   الاول نعمل   
   *from . forms import BookForm*  
   ثم نضيف فى ال dictionary الى كنت عملها لنقل البيانات key جديده و نضع لها value بال class الى عملته فى ال forms  
   *def index(request):  
   context = {  
   ‘category’ : Category.objects.all(),  
   ‘books’ : Book.objects.all(),  
   ‘form’ : BookForm(),  
   }  
   return render(request, ‘pages/index.html’, context)*
6. نذهب الى الصفحه الى فيها ذر ال form الخاص بالاضافه الى هى index  
   نقوم بمسح ال inputs الموجوده   
   نكتب مكنها tag باسم الkey الى كنت عملتها فى function الموجود فى صفحه ال views  
   *{{form}}*
7. اضافه البيانات التى ادخلها المستخدم فى models اى البيانات الخاصه بالموقع  
   نذهب الى ال views و بداخل ال function الخاصه ب الصفحه الى المستخدم بيدخل منها البيانات و هى فى هذه الحاله index  
   نضيف ال if كالتالى  
   if request.method == ‘POST’:  
   add\_book = BookForm(request.POST, request.FILES)  
   if add\_book.is\_valid():  
   add\_book.save()  
   و فى حاله ان ال method تساوى post بنذهب الى ال form الى فى ال index و نضيف تحت tag فتح ال form مباشر tag باسم {%csrf\_token%}
8. نفس الشئ الى عملناه مع ال book هنعمله مع category  
   نبدأ ب forms.py  
   *from . models import Book, Category*  
   class CategoryForm(forms.ModelForm):  
   class Meta:  
   model = Category  
   fields = [‘name’]  
   widgets = {  
   ‘name’ : forms.TextInput(attrs={‘class’:’form-control’})  
   فى ال views.py نعمل import لل CategoryForm ذى الى عملناه لل BookForm او الاصح نضيفه عليها  
   from . forms import BookForm, CategoryForm  
   نذهب الى views.py و نضيف فى ال dictionary الى عملها داخل ال functions علشان نمرر البيانات الى ال return للعرض فى الصفحه key و value لها كالتالى  
   ‘formcat’ : CategoryForm(),  
   مع ملاحظه ان هذا الجذء الذى هو sidebar سوف يعرض فى كل الصفحات لذالك سوف نضيف الكود السابق الى كل ال functions  
   ثم فى ال sidebar.html نقم بالتعديل كالتالى نشيل ال inputs الى هو حطه و نعمل مكانه  
   {% csrf\_token %}  
   ثم  
    {{formcat}}  
    تحت فتحه ال tag الخاصه بال form  
   و فى صفحه ال views داخل ال function نضيف فى ال if الى كنا عملنها قبل كده لل BookForm التالى  
   add\_category = CategoryForm(request.POST)  
   ملحوظه معملناش .FILES علشان مش هنرسل مع البيانات صور  
   if add\_category.is\_valid():  
   add\_category.save()
9. هذه الخطوه اختياريه و هى شئ جمالى لاظهار ال inputs عند الضغط على زر new و ذلك عن طريق ال js   
   ناتى قبل ال {{formcat}} و نعمل tag div و له id  
   <div id=”catinput”>  
   ثم نذهب الى ال base.html نعمل tag ل script ونكتب   
   <script>  
   $(“#catinput”).hide()  
   $(“#addcat”).click(function(){  
   $(“#catinput”).show()  
   });  
   $(“#backcat”).click(function(){  
   $(“#catinput”).hide()  
   });  
   </script>

***####################################***

***الخطوه الثانيه عشر***

***تعديل بيانات الكتاب من المستخدم من خلال صفحه الويب***

***V num #22***

***\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\***

*1 – تعديل ال url الخاصه بالصفحه فى التطبيق بحيث انه يكون لكل كتاب url مختلف  
path(‘update/<int:id>’, views.update, name=’update’),  
نذهب الى الصفحه التى بها زر التعديل فى هذه الحاله هو ال index.html و نذهب الى خانه التعديل و نضع url الخاص بالتعديل فى href كالتالى   
href=”{% url ‘update’ book.id %}”*

*2 – نذهب الى views.py الخاص بالمشروع و نعدل def الخاص ب update كالتالى  
نضيف from Django.shortcuts import redirect, render  
def update(request, id):  
book\_id = Book.objects.get(id=id)  
if request.method == ‘POST’:  
book\_save = BookForm(request.POST, request.FILES, instance = book\_id)*

*If book\_save.is\_valid():  
book\_save.save()  
return redirect(‘/’)  
else: هذا الا else لل if الاولى و ليس الثانيه  
book\_save = BookForm(instance = book\_id)  
بره الا else نكمل  
context = {  
‘form’:book\_save,  
}  
return render(request, ‘pages/update.html’, context)*

*3 – نذهب الى update.html و نقوم بمسح الا input و نضع مكانه  
تحت ال tag بتاعت ال form مباشره نكتب  
{% csrf\_token %}  
جوه ال div الى معاها class باسم card-body  
{{form}}*

***####################################***

***الخطوه الثالثه عشر***

***حذف الكتاب من المستخدم من خلال صفحه الويب***

***V num #23***

***\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\***

*1 – فى ال urls.py بالتطبيق نعدل ال path بتاع ال delete  
path(‘delete/<int:id>’, views.delete, name=’delete’),*

1. *- فى ال index نخش نعمل فى الزر الخاص بالحزف عند href  
   href=”{% url ‘delete’ book.id %}”  
     
   3 – فى صفحه ال views.py نعدل function ال delete كالتالى*

*From Django.shortcuts import redirect, render, get\_object\_or\_404  
ثم   
def delete(request, id):  
book\_delete = get\_object\_or\_404(Book, id=id)  
if request.method == ‘POST’:  
book\_delete.delete()*

*Return redirect(‘/’)*

*نطلع بره ال if   
return render(request, ‘pages/delete.html’)*

1. *فى صفحه ال delete.html  
   تحت tag ال form  
   {% csrf\_token %}  
   بعد كده زر تراجع نشيل منه كلمه Submit و نحط button و نحوله من tag من نوع button الى <a></a> ثم نستبدل data-target=”#” و نحط   
   href=”{% url ‘index’ %}”*

***####################################***

***الخطوه الرابعه عشر***

***تظبيط روابط الموقع***

***V num #24***

***\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\***

*1 – تظبيط روابط التصنيفات فى صفحه ال index*

*فى صفحه ال index.html نذهب الى التصنيفات و نديله class علشان نربط بينه و بين الكتب الى ليها نفس ال category نضيف على tag بتاع ال list class باسم+id كالتالى  
<li class=”nav-item cat{{cat.id}}”>  
و نذهب الى الكتاب و نديله 2 من class  
الاول  
علشان نميز الكتب و نحزفها كلها  
نضيف فى tag ال div بتاع الكتاب class كما هو بعد الخطوه الثانيه   
الثانى   
علشان نرجع الكتب الى ليها نفس category كالتالى   
<div id=”” class=”col-md-4 bookhide book{{book.category.id}}”>  
نذهب الى base.html  
نعمل tag لل script الخاص js  
<script>  
{% for book in books %}  
$(“.cat{{book.category.id}}”).click(function(){  
$(“.bookhide”).hide()  
$(“.book{{book.category.id}}”).show()  
});  
{% endfor %}  
</script>*

*نعمل نفس الشيئ فى ال category الى فى sidebar لاكن سوف نقوم فقط بوضع ال class للزر الى هنضغظ عليه لان باقى الخطوات تم صنعها من قبل فى الخطو السابقة*

*<li style="cursor: pointer;" class="nav-item cat{{cat.id}}">*

*نعمل نفس الشئ فى الكتب التى تم البيع او تاجرها او متاحه  
فى ال sidebar  
نذهب عند كل زر للاختيار  
عند كل اختيار نحط class  
statussold  
statusrental  
statusavailble*

*ثم نذهب الى index حيث يظهر التغيرات فى الكتب و نضيف على الكتاب class  
book{{book.status}}  
ثم فى ال base.html  
$(“.status{{book.status}}”).click(function(){*

*$(“.bookhide”).hide()  
$(“.book{{book.status}}”).show()  
});*

تعديل ال home , nouvil, abdelrahman, newbooks  
كل الى بنعمله بنعدلhref لكل واحده ب tag بتاع url كالتالى   
“{% url ‘index’ %}”