

python ဖြင့် web လောကထဲ ခြေစုံပစ်ဝင်ခြင်း

သင်ခန်းစာ 1. django website တစ်ခု လွှင့်တင်ခြင်း

1 . django ကို အောက်က command နဲ့ terminal မှာရိုက်ပြီး install လုပ်ပါ။

```
pip install django
```

django ကို install လုပ်ပါလိမ့်မယ်။

=====

2 . folder တစ်ခုဆောက်ဖို့ အောက်ကလို ထပ်ရိုက်ပါ။

```
django-admin startproject shop .
```

shop နေရာမှာ ကြိုက်တဲ့အမည်ပေးနိုင်ပါတယ်။ full stop ထည့်ရပါလိမ့်မယ်။ (shop space . space)

```
shop
_ init.py
_ setings.py
_ urls.py
_ wsgi.py
```

```
manage.py
```

အပေါ်ကလို ဖိုင်တွေ ဆောက်ပေးပါလိမ့်မယ်။

=====

3 . virtual environment တစ်ခုတည်ဆောက်ပြီး website ကို စမ်းဖို့ virtual env ကို install လုပ်ပါ။

```
pip install virtualenv
```

=====

1. virtual environment လည်းတည်ဆောက်ပြီးပြီ။

2. django လည်း install လုပ်ပြီးပြီ။

3. website အတွက် လိုအပ်တဲ့ အခြေခံဖိုင်နဲ့ code တွေလည်းရပြီ။

အပေါ်သုံးချက်ရပြီဆိုတော့ manage.py ဆိုတဲ့ဖိုင်ကို server မှာ run ပါမယ်။

အောက်က အတိုင်း terminal မှာ ရိုက်ပါ။

```
python manage.py runserver
```

စာတွေအများကြီးထဲမှာ အောက်က လင့်ကို တွေ့ပါလိမ့်မယ်။

<http://127.0.0.1:8000/>

.....

=====

လင့်ကိုနှိပ်လိုက်ရင် ပုံထဲကအတိုင်းပြပါလိမ့်မယ်။

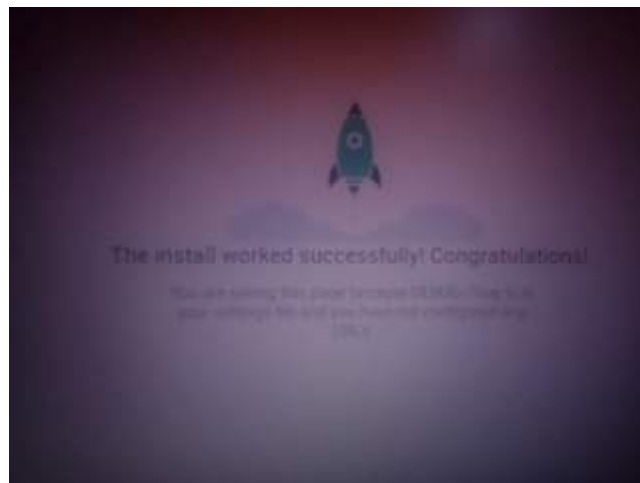
===== The End =====

ဒီလိုရေးဖို့ လေးနာရီလောက် စမ်းလိုက်ရပါတယ်။

တကယ်ဆိုရင် အရင်လေးလလောက်ထဲက စမ်းမရလို့ ပစ်ထားတာပါ။

ကျွန်တော်က အဲ့အပိုင်းတွေကို တော်တော်လေးအားနည်းပါတယ်။

ရှေ့ဆက်ရေးမယ့်ဟာတွေကတော့ ကျွန်တော်ရပြီးသားတွေမို့ သင်တာမြန်ပါလိမ့်မယ်။



python ဖြင့် web လောကထဲ ခြေစုံပစ်ဝင်ခြင်း

သင်ခန်းစာ 2.

django website တွင် ထုတ်ကုန်များ ထပ်ထည့်ခြင်း

ထုတ်ကုန်များကို link ချိတ်ပြီး ပြသခြင်း

အရင်သင်ခန်းစာမှာ shop ဆိုတဲ့ project တစ်ခုဆောက်ခဲ့ပါတယ်။

သူ့ကို internal storage ထဲမှာ ရှာကြည့်ပါ။

အဲ့ထဲက wsgi.py ဆိုတဲ့ဖိုင်မှာ ပုံတစ်ကအတိုင်း No.18, 19, 20 သုံးကြောင်းထပ်ထည့်ပါ။

```
wsgi.py
/storage/emulate...

new* new* new* new* new*

3
4 It exposes the WSGI callable as a module-level
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/3.0/howt
8
9
10 import os
11
12 from django.core.wsgi import get_wsgi_appli
13
14 os.environ.setdefault('DJANGO_SETTINGS_M
15
16 application = get_wsgi_application()
17
18 from django.http import HttpResponse
19 def x(request):
20     return HttpResponse('Apples')
```

urls.py ဖိုင်မှာ ပုံနှစ်ကလို No.18 နဲ့ 21 ထပ်ထည့်ပါ။

```
urls.py
/storage/emulate...

new* new* new* new* new*

5 Examples:
6 Function views
7     1. Add an import: from my_app import vi
8     2. Add a URL to urlpatterns: path("", views
9 Class-based views
10    1. Add an import: from other_app.views i
11    2. Add a URL to urlpatterns: path("", Home
12 Including another URLconf
13    1. Import the include() function: from djar
14    2. Add a URL to urlpatterns: path('blog/', i
15
16 from django.contrib import admin
17 from django.urls import path
18 from . wsgi import x
19 urlpatterns = [
20     path('admin/', admin.site.urls),
21     path('product/', x)
22 ]
23
```

ပြီးရင် server ကို ပြန် run ပါမယ်။ပုံသုံးကလို ပေါ်လာရင် 127.0.0.1:8000 ဆိုတဲ့စာကိုနှိပ်ပြီး website ကိုဖွင့်ပါ။

```
/storage/emulated/0 $ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 17 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.

January 27, 2020 - 23:33:43
Django version 3.0.2, using settings 'shop.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
Not Found: /
```

ပုံလေးကလို product လို့ထပ်ထည့်ပါ။



enter နှိပ်ရင် ပန်သီးဆိုတဲ့ product ကျလာပါလိမ့်မယ်။



လှလှပပပြချင်ရင်တော့ CSS ထပ်ထည့်ရပါလိမ့်မယ်။

နောက်မှ vedio တင်ပေးပါမယ်။

python ဖြင့် web လောကထဲ ခြေစုံပစ်ဝင်ခြင်း

သင်ခန်းစာ 3. djangoဖြင့် website တွင် admin အဖြစ် login ပြုလုပ်ခြင်းကို ရေးသားခြင်း

1. terminal မှာ အောက်ကလိုရိုက်ပါ။
python manage.py migrate

ပုံတစ်ကလို ပြပါလိမ့်မယ်။

```
/storage/emulated/0 $ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying sessions.0001_initial... OK
```

2. Super user ဖန်တီးပါ။

python manage.py createsuperuser

username တောင်းရင် နာမည်တစ်ခုပေးပါ။
ပုံနှစ်ကို ကြည့်ပါ။ ကျွန်တော်က admin လို့ ပေးပါတယ်။

```
/storage/emulated/0 $ python manage.py createsuperuser
Username (leave blank to use 'u0_a104'): admin
```

email တောင်းရင် ကိုယ့်အကောင့်ထည့်ပါ။
ပုံသုံးကို ကြည့်ပါ။

```
/storage/emulated/0 $ python manage.py createsuperuser
Username (leave blank to use 'u0_a104'): admin
Email address: zunmekhaing11@gmail.com
Password: 
```

password တောင်းရင် ကြိုက်နှစ်သက်ရာထည့်ပါ။
password ရိုက်ရင် စာမပေါ်ပါဘူး။
ကျွန်တော်က y လို့ထည့်ပါတယ်။

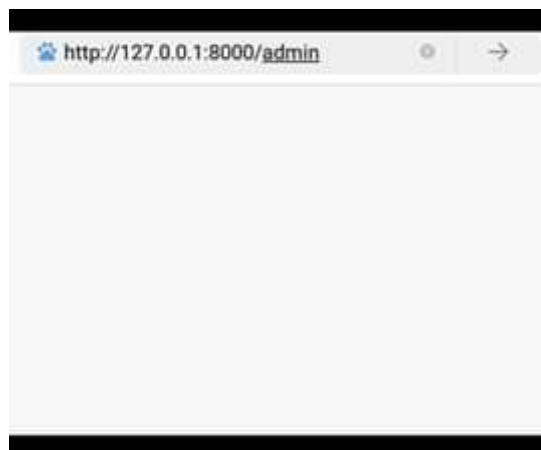
password ကို ထပ်ရိုက်ခိုင်းရင် အရင် password ကို ပြန်ထည့်ပါ။

password ကို y လို့ တစ်လုံးတည်းပေးထားလို့ လုံခြုံမှုမရှိဘူးလို့ပြပါတယ်။ အတည်ပြုမှာလားလို့ မေးပါတယ်။
အတည်ပြုမှာဆိုရင် y ကို ရေးခိုင်းလို့ y လို့ ရေးပါတယ်။

super user အဖြစ် မှတ်ပုံတင်ပြီးပါပြီ။
ပုံလေးကို ကြည့်ပါ။

```
python manage.py createsuperuser
Username (leave blank to use 'root'): admin
Email address: zunwekhaing11@gmail.com
Password:
Password (again):
Error: Your passwords didn't match.
Password:
Password (again):
Error: Your passwords didn't match.
Password:
Password (again):
Error: Your passwords didn't match.
Password:
Error: This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
/storage/emulated/0 $
```

3. server ကို runပါ။
python manage.py runserver
link ကိုနှိပ်ပါ။
4. ပုံငါးကလို admin ကိုထပ်ထည့်ပါ။



Enter နှိပ်ရင် ပုံခြောက်လို ပြပါမယ်။

အရင်ထည့်ခဲ့တဲ့ username နဲ့ password ကို ထည့်ပါ။
(admin နဲ့ y ကိုထည့်ပါတယ်။)

5. ပုံ ခုနစ် လိုပြပါမယ်။
admin တစ်ယောက်ဖြစ်သွားပါပြီ။

===== End =====

နက်ဖြန်တော့ application ထည့်တာပေါ့။

python ဖြင့် web လောကထဲ ခြေစုံပစ်ဝင်ခြင်း

သင်ခန်းစာ 4. Website admin (ပိုင်ရှင်) အသုံးပြုရန် / ရောင်းကုန်များ ထည့်နိုင်ရန် application တစ်ခုဖန်တီးခြင်း

သင်ခန်းစာနှစ်မှာ ရောင်းမယ့်ပစ္စည်းတွေကို website မှာ ပြတဲ့နည်းကို သင်ခဲ့ပါတယ်။
website သုံးမယ့်ပိုင်ရှင်က အဲ့လိုမရေးတတ်ပါဘူး။
သူ့အတွက် application တစ်ခုရေးပေးရပါလိမ့်မယ်။

သင်ခန်းစာ 3 အထိလုပ်ခဲ့သမျှဖိုင်တွေကို အကုန်ပြန်ဖျက်ပါ။manage.py ဖိုင်ကိုလည်း ရှာပြီးဖျက်ပါ။
ဘာကြောင့်ဖျက်ရလဲဆိုတော့ သင်ခန်းစာ 3 ထိက အခြေခံနားလည်အောင်ပြတာမို့ပါ။သုံးလေးခါလောက် ဖျက်လိုက်ပြန်စလိုက်လုပ်ပြီး လေ့ကျင့်ပါ။

လေ့ကျင့်ပြီးလို့ အခြေခံရပြီဆိုရင် အစအဆုံးပြန်တည်ဆောက်ပြပါမယ်။

step.1

django နဲ့ virtualenv ကို install လုပ်ပြီးသားမို့ ထပ်လုပ်စရာ မလိုတော့ပါဘူး။
(pip install django နဲ့ pip install virtualenv ပါ)

step. 2

django-admin startproject Shop .

(space . space ကို မမေ့ပါနဲ့။)

ဒါဆိုရင် website တစ်ခု အတွက် project ဖိုင်တွဲဆောက်ပေးပါလိမ့်မယ်။

step. 3

python manage.py runserver

step. 4

terminal ကနေထွက်ပြီး ပြန်ဝင်ပါ။

python manage.py migrate

step. 5

python manage.py createsuperuser

user name နဲ့ password ထည့်ပါ။

အတည်ပြုမှာလားမေးရင် y လို့ထည့်ပါ။

step. 6

python manage.py startapp product

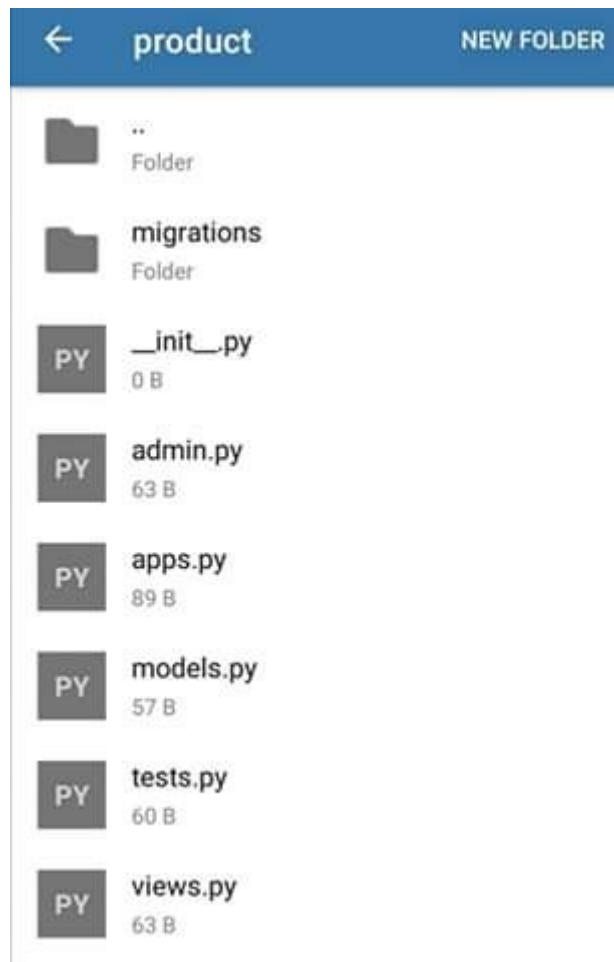
ဒါဆိုရင် product လို့အမည်ပေးထားတဲ့ application ဖိုင်ကို ဆောက်ပေးပါလိမ့်မယ်။

product နောက်မှာ . မထည့်ရပါဘူး။

step. 7

product ဆိုတဲ့ဖိုင်တွဲကို ရှာပြီးဖွင့်ပါ။

ပုံတစ်လို့ပြပါမယ်။



models.py ဖိုင်ကို ဖွင့်ပါ။
 ပုံနှစ်ကလို မွေးစား class တစ်ခု ဆောက်ပါ။
 class အမည်ကို ကျွန်တော်ပေးတဲ့အတိုင်းပဲပေးပါ။
 ဖိုင်ကို save ပါ။
 (ဒါဟာ application ရဲ့ ရုပ်ထွက်ကို ရေးလိုက်တာပါ။)

```

models.py
/storage/emulate...

1 from django.db import models
2
3 # Create your models here.
4
5 class product_model(models.Model):
6     name = models.CharField(max_length=255)
7     price = models.FloatField()
8     stock = models.IntegerField()
9     image = models.CharField(max_length=255)
10

```

ရုပ်ထွက်ပိုင်းကို admin site မှာ ပြအောင်လို့ register လုပ်ရပါမယ်။admin.py ဖိုင်ကို ဖွင့်ပြီး အောက်ကနစ်ကြောင်းကို ထပ်ဖြည့်ပါ။

```

from .models import product_model
admin.site.register(product_model)
(models ဆိုတဲ့ package ဟာ ကျွန်တော်တို့ရေးထားတာပါ။
django ရဲ့ package နဲ့နာမည်တူနေလို့ import လုပ်ရင် .models လို့ ရေးရပါတယ်။
. က လက်ရှိခိုင်းတဲ့ထဲက package ကို ရည်ညွှန်းတာပါ။ )

```

step. 8
product ဖိုင်တွဲထဲက apps.py ကို ရှာပြီးဖွင့်ပါ။
ပုံသုံးကလို ပြပါမယ်။
product ဆိုတဲ့ application ကို အလုပ်လုပ်စေချင်ရင် ProductConfig ကို ခေါ်ရမယ်ဆို
တာလေးကို မှတ်ထားပါ။



```

1 from django.apps import AppConfig
2
3
4 class ProductConfig(AppConfig):
5     name = 'product'
6

```

step. 9
Shop ဆိုတဲ့ folder ကိုဖွင့်ပါ။(ဒါက Website ရဲ့ ပင်မခိုင်းတဲ့ပါ။)
setting.py ကို ဖွင့်ပြီး ပုံလေးကလိုမျိုး
No.39 မှာ , ထည့်ပါ။
No. 40 မှာ 'product.apps.ProductConfig' ကိုထည့်ပါ။



```

10 # This file contains Django's project configuration settings.
11
12
13 import os
14
15 # Build paths inside the project like this: os.path.join(BASE_DIR, ...)
16 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
17
18
19 # Quick-start development settings - unsuitable for production
20 # See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/
21
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = 'qontl*5&)x++hvch89t8#+s1jy7q3r6gl%1d+5a33it-x*wgqf'
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'product.apps.ProductConfig'
41 ]
42

```

(product application ကို Website မှာ install လုပ်ဖို့
product package ထဲက apps module ထဲမှာရှိတဲ့ ProductConfig ဆိုတဲ့ class ကို
ထည့်တာပါ။
product package က ကျွန်တော်တို့ရေးထားတဲ့ product application ပါ။)

step. 10

terminal ကို ပြန်သွားပြီး အောက်က command ကိုရိုက်ရင် product မိုင်တွဲထဲက migration မိုင်ထဲမှာ 0001_initial.py ဆိုတဲ့ မိုင်ကို ဆောက်ပေးပါလိမ့်မယ်။
ပုံငါးကို ကြည့်ပါ။



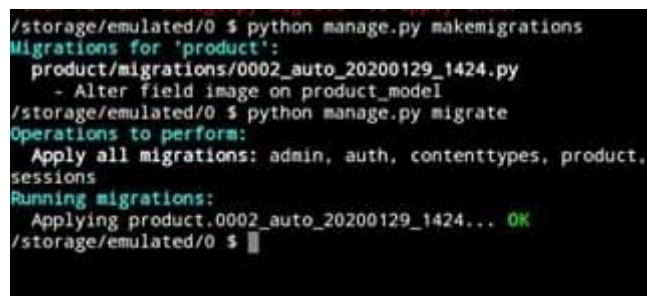
```
1 # Generated by Django 3.0.2 on 2020-01-29 07:21
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     initial = True
9
10    dependencies = [
11    ]
12
13    operations = [
14        migrations.CreateModel(
15            name='product_model',
16            fields=[
17                ('id', models.AutoField(auto_created=True, primary_key=True, serial
18                ('name', models.CharField(max_length=255)),
19                ('price', models.FloatField()),
20                ('stock', models.IntegerField()),
21                ('image', models.CharField(max_length=2)),
22            ],
23        ),
24    ]
```

python manage.py makemigrations

step. 11

အောက်က command ကို ထပ်ရိုက်ရင် product ဆိုတဲ့ application ဟာ ပင်မ Website ထဲမှာ install လုပ်ပြီးသွားပါပြီ။ပုံခြောက်ကို ကြည့်ပါ။

python manage.py migrate



```
/storage/emulated/0 $ python manage.py makemigrations
Migrations for 'product':
  product/migrations/0002_auto_20200129_1424.py
    - Alter field image on product_model
/storage/emulated/0 $ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, product,
sessions
Running migrations:
  Applying product.0002_auto_20200129_1424... OK
/storage/emulated/0 $
```

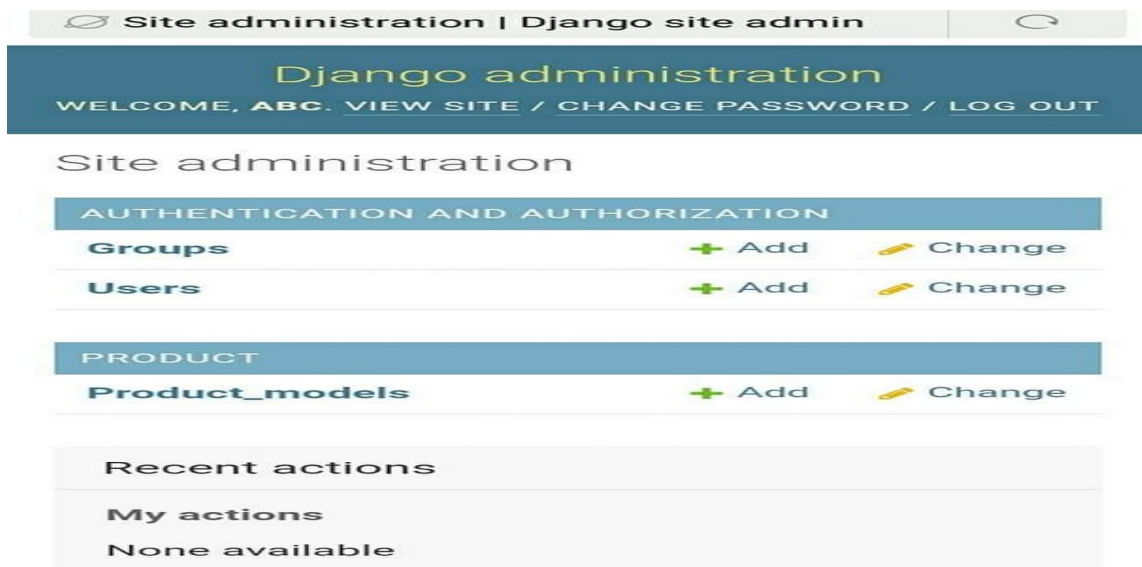
step. 12

python manage.py runserver လို့ ရိုက်ပြီး link ကို ဝင်ပါ။

ပုံခုနစ်အတိုင်း link နောက်မှာ admin လို့ ထပ်ထည့်ပြီး enter နှိပ်ပါ။



user_name နဲ့ password ထည့်ရင် admin နေရာကို ရောက်ပါမယ်။
ပုံရိပ်ကိုကြည့်ပါ။



step. 13

ပုံရိပ်မှာ Product ဆိုတဲ့ application ရဲ့ add ခလုတ်ကို နှိပ်ပါ။

ပုံကိုးကလို ဖြည့်ပြီး save ပါ။

image နေရာမှ ဖြည့်ဖို့ link ကို comment ကနေ copy ယူပါ။

ဒါဆိုရင်တော့ မိမိထုတ်ကုန်တွေကို website ပေါ်တင်ဖို့ application တစ်ခု ရသွားပါပြီ။

လိုအပ်တဲ့ database နဲ့ တစ်ခြားအရာတွေကို django က သူ့ဘာသာ ဆောက်ပေးသွားတယ်လို့ မှတ်ထားပါ။

Add product_model

Name:

Price:

Stock:

Image:

SAVE

Image link..

<https://www.shutterstock.com/image-photo/yellow-green-red-apples-isolated-on-207713410>

===== END =====

ဝယ်သူတွေကို ဘယ်လိုပြမလဲဆိုတာက နောက်သင်ခန်းစာမှာပါပဲမယ်။

python ဖြင့် web လောကထဲ ခြေစုံပစ်ဝင်ခြင်း

သင်ခန်းစာ 5. အရောင်းပစ္စည်းစာရင်းကို ပိုင်ရှင်ကြည့်နိုင်အောင် table ဖြင့် ပြသစေခြင်း

အရင်သင်ခန်းစာမှာ ပုံတစ်ကလိုမျိုးရေးပြီး အရောင်းပစ္စည်းတင်ဖို့ ရုပ်ထွက်ကို ဖန်တီးခဲ့ပါတယ်။

ပိုင်ရှင်တင်လိုက်တဲ့ အရောင်းပစ္စည်းရဲ့ name price stock စတာတွေရဲ့တန်ဖိုးကို ယူပြီး ပိုင်ရှင် ဝန်ကြည့်ဖို့ table ဆောက်ပါမယ်။

step.1

ပုံနှစ်ကလို admin.py မှာ မွေးစား class တစ်ခုဆောက်ပြီး register လုပ်လိုက်ပါ။
ဒါဆိုရင် table ဖြစ်ပါပြီ။
မိုင်ကို save ပါ။



step.2

ရေးချင်တာရေးပြီးတိုင်း manage.py မှာ မိုင်ကို migrate လုပ်ရပါမယ်။

မလုပ်လည်းရတာတွေရှိပေမယ့် အကျင့်ဖြစ်အောင်လုပ်ပါ။

python manage.py makemigrations

python manage.py migrate

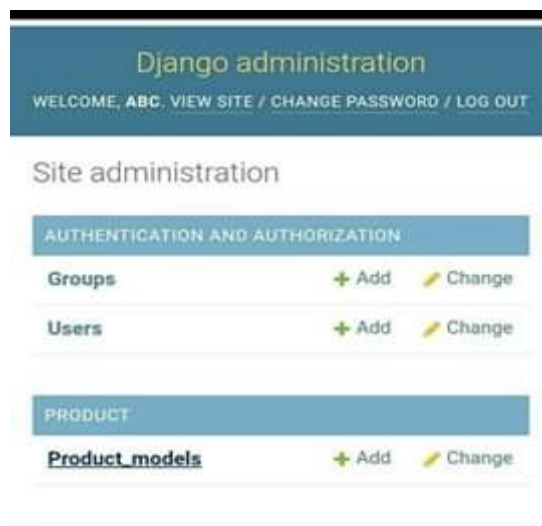
step.3

အသစ်ဖြည့်ပြီးတိုင်း server ကို run ပြီး စစ်ကြည့်ဖို့လိုပါတယ်။

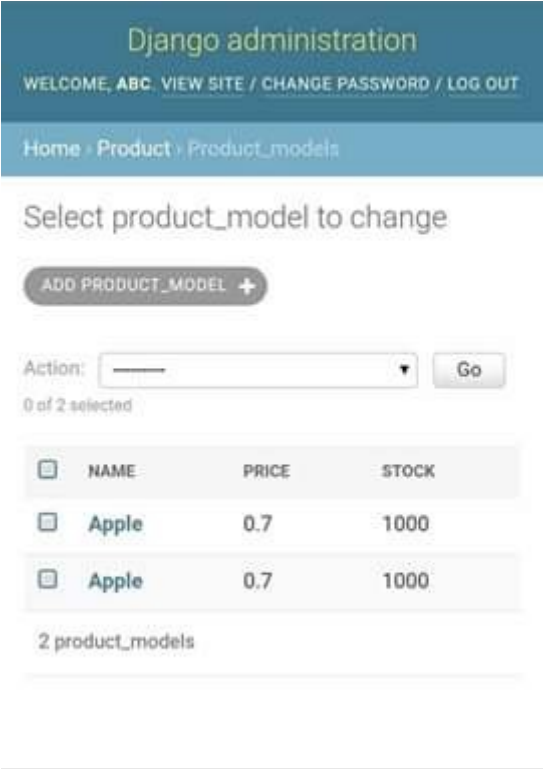
python manage.py runserver

link ကို ဝင်ပြီး admin ထပ်ထည့်ပါ။

ပုံသုံးကလို ပေါ်လာရင် product application ရဲ့ Product_models ခလုတ်ကိုနှိပ်ပါ။



ပုံလေးကလို အရင်save ထားတဲ့ ပစ္စည်းတွေကို table နဲ့ ပြပါလိမ့်မယ်။



add ခုလုတ်နဲ့ ပစ္စည်းတွေ ထပ်ဖြည့်ရင် table မှာလည်း အသစ်တိုးတာကို ပြပါလိမ့်မယ်။

===== END =====

python ဖြင့် web လောကထဲ ခြေစုံပစ်ဝင်ခြင်း

သင်ခန်းစာ . 6

product application တွင် ဝယ်ယူအားပြသစေသော လုပ်ဆောင်ချက်ထပ်ထည့်ခြင်း

step.1 (ပုံတစ်)

product ဖိုင်တွဲထဲက views.py ဖိုင်မှာ No. 3,5,6 သုံးကြောင်း ထည့်ပါ။

(Products တွေကို website မှာပြဖို့ product application မှာ function တစ်ခုရေးတာပါ။)

```
1 from django.shortcuts import render
2
3 from django.http import HttpResponseRedirect
4
5 def x(request):
6     return HttpResponseRedirect('Products')
7
8 # Create your views here.
9
```

=====

step.2 (ပုံနှစ်)

shop ဆိုတဲ့ ပင်မဖိုင်တွဲရဲ့ urls.py ဖိုင်ကို ဖွင့်ပြီး No. 18 နဲ့ 22 ကိုထည့်ပါ။

21 မှာ , ထည့်ပါ။

(product application ရဲ့ views.py ဖိုင်ထဲက x function ကို products ဆိုတဲ့ link နဲ့ ချိတ်လိုက်တာပါ။)

```
1 Shop URL Configuration
2
3 The 'urlpatterns' list routes URLs to views. For more information please see
4 https://docs.djangoproject.com/en/3.0/topics/http/urls/
5 Examples:
6 Function views
7 1. Add an import: from my_app import views
8 2. Add a URL to urlpatterns: path('', views.home, name='home')
9 Class-based views
10 1. Add an import: from other_app.views import Home
11 2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13 1. Import the include() function: from django.urls import include, path
14 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15
16 from django.contrib import admin
17 from django.urls import path
18 from product.views import x
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('products/', x)
23 ]
24
```

=====

step. 3. (ပုံသုံး)

Website ကို ပြန်ဖွင့်ပြီး မူလလင့်ခ်နောက်မှာ ကျွန်တော်တို့ချိတ်ထားတဲ့ products ထပ်ထည့်လိုက်ရင် x function ရဲ့လုပ်ဆောင်ချက်ကို ပြပါလိမ့်မယ်။



=====

step. 4. (ပုံလေးကို comment တွင်ကြည့်ပါ။)



x function နဲ့ html တွေ ချိတ်လိုက်ရင် အထွေထွေဗဟုသုတ .2 ကလို ဖြစ်ပါလိမ့်မယ်။

ပုံတစ် No.6 ရဲ့ "Products" နေရာမှာ html ထည့်လိုက်ပါ။

comment မှာ copy ယူပါ။

```
from django.shortcuts import render
```

```
from django.http import HttpResponse
```

```
def x(request):
```

```
    return HttpResponse("""<h1> products </h1>
```

```

```

 Apples (\$ [0.7](#))

 Orange (\$ [0.7](#)) """)

(သင်ခန်းစာ နှစ်နဲ့ သဘောတရားတူပါတယ်။)

python ဖြင့် web လောကထဲ ခြေစုံပစ်ဝင်ခြင်း

သင်ခန်းစာ . 7 ပိုင်ရှင်တင်သော ပစ္စည်းများကို products link တွင် ချိတ်ဆက်ပြသခြင်း

Step.1 (ပုံတစ်)

product application မှာ models.py မှာ ဝင်ပြီး image ဆိုတဲ့ variable ရဲ့ maximum length ကို 255 ကနေ 355 လို့ ပြောင်းပါ။

ပုံအတွက် link က စာလုံး 255 ကျော်နေလို့ ပြင်လိုက်တာပါ။

=====

Step. 2

product application မှာ views.py မှာ ဝင်ပြီး ပုံနှစ်အတိုင်းရေးပါ။ (မန့်မှာ copy ယူပါ။)

(admin ထည့်သမျှ product data တွေကို ယူပြီး html မှာ ဖော်ပြပါ။)

```
from django.shortcuts import render
from .models import product_model
```

```
def x(request):
    products=product_model.objects.all()
    retrun render(request,"x.html",{ "products":products})
```

=====

Step. 3

product application ဖိုင်တွဲထဲမှာ templates ဆိုတဲ့ ဖိုင်တွဲတစ်ခု ထပ်ဆောက်ပါ။

templates ဖိုင်တွဲထဲမှာ x.html ဆိုတဲ့ ဖိုင်တစ်ခုဆောက်ပြီး ပုံသုံးကလိုရေးပါ။

(မန့်မှာ copy ယူပါ။)

(html ထဲကို python နဲ့ ဝင်ပြီး ဖြတ်ညှပ်ကပ်လုပ်တာပါ။

ခိုင်းစားတယ်ပေါ့ဗျာ။)



```
<h1> Products </h1>
<ul> {% for product in products %}
    <img src={{product.image}}
    style="width:50:">
        <li> {{ product.name}} (${{product.price}}) </li>
        <li> stock( {{ product.stock }} )</li>
    {% endfor %}
</ul>
```

=====

Step. 4

server ရဲ့ admin control site ကို ဝင်ပြီး အရင် save ထားတဲ့ products တွေကို ဖျက်ပါ။ image link တွေမှားနေလို့ ဖျက်ခိုင်းတာပါ။

Apple , Orange, Banana ဆိုပြီး အသစ်သုံးခုထည့်ပါ။

image နေရာမှာ ထည့်ရမယ့် link တွေကို comment မှာ ယူပါ။

Apple image link

https://s.yimg.com/lo/api/res/1.2/M01hB0GCoFajAqXVaIm2gw--~B/YXBwaWQ9eWlzZWFiY2g7Zmk9Zml0O2dIPTAwNjYwMDtncz0wMEEzMdA7aD00MDA7dz03NjI-/https://greatist.com/sites/default/files/styles/fb-1200x630/public/wp-content/uploads/2011/11/Apples.jpg?itok=Cqn_I8Ms.cf.jpg

Orange image link

<https://s.yimg.com/lo/api/res/1.2/1fTROiUFSQP1XVaocgNj2Q--~B/YXBwaWQ9eWlzZWFiY2g7Zmk9Zml0O2dIPTAwNjYwMDtncz0wMEEzMdA7aD00MDA7dz02MDA-/https://zkanakhazana.files.wordpress.com/2012/01/orange.jpg.cf.jpg>

Banan image link

https://s.yimg.com/lo/api/res/1.2/DoPv9CYI_96_s18CZqwSfQ--~B/YXBwaWQ9eWlzZWFiY2g7Zmk9Zml0O2dIPTAwNjYwMDtncz0wMEEzMDA7aD00MDA7dz02MDE-/http://img.aws.livestrongcdn.com/ls-article-image-640/ds-photo/getty/article/106/160/155251298.jpg.cf.jpg

=====

Step. 5
products link ကို ဝင်ရင် ပုံလေးကလို့ ပြပါလိမ့်မယ်။

===== End =====



python ဖြင့် web လောကထဲ ခြေစုံပစ်ဝင်ခြင်း

သင်ခန်းစာ . 8. စတိုင်ကျအောင် cascading style sheets (CSS) သုံးခြင်း

Step.1

product folder ထဲက template folder ကိုဖွင့်ပြီး x.html file ထဲက code ကို comment ထဲက code နဲ့အစားထိုးပါ။

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport"
      content="width=device-width,initial-scale=1">
<style>.card{ box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
      transition: 0.3s;
      width: 40%; }
      .card:hover { box-shadoow: 0 8px 16px 0 rgba(0,0,0,0.2);}
      .container {padding: 2px 16px;}
</style>
</head>

<body> {% for product in products %}
<h2> {{product.name}}</h2>
<div class="card">
    <img src={{product.image}}
    style="width:100%">
    <div class="container">
        <h4><b>price(${{product.price}}</b></h4>
        <h4><b>Instock( {{product.stock}}</b></h4>
        <p>About apples ... .. </p>
        <div class="card-action">
            <a href="#">Order</a>
        </div>
    </div>
</div>
</body>
</html>
{% endfor %}
```

=====

Step. 2

products link ကို ဝင်ကြည့်ရင် ပုံထဲကလို လှလှပပပြပါလိမ့်မယ်။

လုပ်/မလုပ်သိအောင် လက်သည်းကွင်းကျန်နေတာကို ကိုယ့်ဘာသာ ဖြည့်ပြီး ပြပါ။



=====

Django နမူနာသင်ခန်းစာပြီးပါပြီ။

အခြေခံကိုတော့ ဒီလောက်ဆိုရပါပြီ။

အားတဲ့ရက်မှာ အစအဆုံး vedio လေးတင်ပေးပါ့မယ်။

ကိုယ့်ဘာသာ website ရေးတတ်ပြီဆိုရင် လက်တွေ့လေ့ကျင့်ခြင်း သင်ခန်းစာမှာ သူများရေးတာတွေကို တုပြီး ရေးပြပါမယ်။

စေတနာရှင်များရှိရင် django 1 to 8 ကို pdf လေး ထုတ်ပေးကြပါဦး။

python ဖြင့် web လောကထဲ ခြေစုံပစ်ဝင်ခြင်း

သင်ခန်းစာ 9. dynamic web page တစ်ခု စမ်းသပ်ခြင်း

dynamic ဆိုတာက ပြောင်းလွယ်ပြင်လွယ်ရှိတာကို ပြောတာပါ။

dynamic web page ကတော့ အမြဲတမ်း update ဖြစ်နေမယ့် web စာမျက်နှာပေါ့။

ရိုးရိုးလေးနဲ့ dynamic ဖြစ်အောင် စမ်းကြည့်ပါမယ်။

step.1

ပင်မ ဖိုင်တွဲဖြစ်တဲ့ Shop ဆိုတဲ့ folder မှာ views.py ဆိုတဲ့ ဖိုင်တစ်ခု အသစ်ဆောက်ပြီး ပုံတစ်ကလို ထည့်ပါ။

(code ကို မနီမှာ copy ယူပါ။)



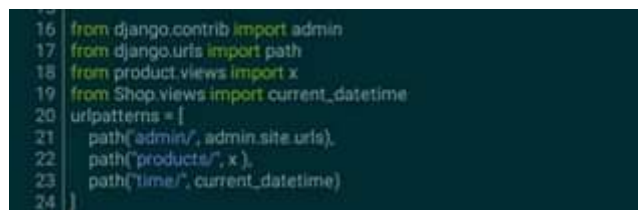
```
1 from django.http import HttpResponse
2 import datetime
3 def current_datetime(request):
4     now = datetime.datetime.now()
5     html = "<html><body><h1>It is now</h1><i>%s.</body></html>" % now
6     return HttpResponse(html)
7
```

```
from django.http import HttpResponse
import datetime
```

```
def current_datetime(request):
    now=datetime.datetime.now()
    html="<html><body><h1>It is now </h1><i>%s.</body></html>" %
    now
    return HttpResponse(html)
```

step.2

ပင်မဖိုင်တွဲ Shop ရဲ့ urls.py ဖိုင်မှာ ပုံနှစ်ကလို No.19, 23 နှစ်ကြောင်းထပ်ထည့်ပါ။
(browser မှာ time လို့ ရိုက်ရှာရင် ပြအောင်လို့ပါ။)



```
16 from django.contrib import admin
17 from django.urls import path
18 from product.views import x
19 from Shop.views import current_datetime
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('products/', x),
23     path('time/', current_datetime)
24 ]
```

```
from django.contrib import admin
from django.urls import path
from product.views import x
from Shop.views import current_datetime
```

```
urlpatterns= [
    path('admin/', admin.site.urls),
```



```
path("produsts/",x)
path("time/", current_datetime)
```

step.3

server run ပြီး time ဆိုတဲ့ link ကို သွားရင် ပုံသုံးကလို လက်ရှိအချိန်ကို ပြပါလိမ့်မယ်။
browser ကို refresh လုပ်ရင် လက်ရှိအချိန်ကို up date လုပ်ပြီးပြပါလိမ့်မယ်။
တစ်ရေးနီးထကြည့်လည်း update ဖြစ်နေပါလိမ့်မယ်။



(date time module အကြောင်းကို မသိရင် တင်ပြီးသားရှာဖတ်ပါ။)

python ဖြင့် web လောကထဲ ခြေစုံပစ်ဝင်ခြင်း

သင်ခန်းစာ 10. ပစ္စည်းအသစ်များ အလွယ်တကူ ကြည့်နိုင်ရန် Web page တစ်ခုထပ်ထည့်ခြင်း

ရှေ့သင်ခန်းစာတွေမှာ admin က ပစ္စည်းအရောင်းပိုင်စွဲတွေကို တင်တဲ့အခါ product ဆိုတဲ့ link မှာ အစဉ်လိုက်ပြပါတယ်။

ဝယ်နေကြဖောက်သည်တွေက ပစ္စည်းအသစ်တွေကိုပဲ အရင် ကြည့်ချင်ပါလိမ့်မယ်။

ဒါကြောင့် နောက်ဆုံးတင်တဲ့ ပစ္စည်းကို အရင်ကြည့်နိုင်အောင် new ဆိုတဲ့ web page တစ်ခုနဲ့ ပြပါမယ်။

=====

step.1

product ဖိုင်တွဲထဲက views.py ဖိုင်မှာ new ဆိုတဲ့ function တစ်ခုရေးပါ။ (ပုံတစ်) (စာရင်းတစ်ခုကို နောက်ဆုံးကပြဖို့ [စမှတ် : ဆုံးမှတ် : -1] ဆိုတာကို သုံးထားပါတယ်။) code ကို comment မှာ copy ယူပါ။

```
def new  
(request):  
    new =
```



```
1 from django.shortcuts import render  
2 from .models import product_model  
3  
4 def x(request):  
5     products = product_model.objects.all()  
6     return render(request, "x.html", { "products": products })  
7  
8  
9 def new(request):  
10     new = product_model.objects.all()[::-1]  
11     return render(request, "x.html", { "products": new })  
12
```

```
product_model.object.all()[::-1]  
    return render(request, "x.html", {"products":new})
```

=====

step.2


Shop ဖိုင်တွဲထဲက urls.py ဖိုင်ထဲမှာ

No.18 မှာ new function ကို import လုပ်ပြီး

No.24 ကလို new ဆိုတဲ့ link အသစ်ထပ်ထည့်ပြီး new function နဲ့ချိတ်ပါ။

ပုံနှစ်ကိုကြည့်ပါ။

code ကို comment မှာ copy ယူပါ။



```
15  
16 from django.contrib import admin  
17 from django.urls import path  
18 from product.views import x, new  
19 from Shop.views import current_datetime  
20  
21 urlpatterns = [  
22     path('admin/', admin.site.urls),  
23     path('products/', x),  
24     path('new/', new),  
25     path('time/', current_datetime)  
26 ]  
27
```

```

from dhango.confrib import admin
from django.uls import path
from product.views import x, new
from Shop.views import current_datetime

urlpatterns=[
    path('admin/',admin.site.urls),
    path("products/",x),
    path("new/",new),
    path("time/",current_datetime)
]

```

=====

step.3

new ဆိုတဲ့ link ကို သွားရင် ပုံသုံးကလိုမျိုး နောက်ဆုံး update လုပ်ထားတဲ့ ပစ္စည်းကို အရင် ပေါ်လိမ့်မယ်။

(products link ကို သွားရင်တော့ ပထမဆုံးတင်တာကို စပြီးပြပါလိမ့်မယ်။)

