Django

1. Create <project folder> and enter the folder
2. Virtual environment
   1. virtualenv venv
   2. virtualenv venv -p python3
3. Activate
   1. source venv/bin/activate
4. Make directory src then enter to src
   1. mkdir src
   2. cd src
5. Install django and create project
   1. pip install django==1.11.15
   2. pip install pillow psycopg2 ( psycopg2-binary )
   3. django-admin.py startproject <project folder name>
6. Change dir to <project folder name>
   1. cd <project folder name>
   2. mkdir static media templates
   3. python manage.py runserver
7. To activate virtualenv
   1. source ../../venv/bin/activate
8. To open python shell
   1. python manage.py shell
9. Create a new App
   1. python manage.py startapp <app name>
10. Create database and user in phppgadmin
    1. sudo su postgres
    2. createdb <dbname>
    3. createuser shibil -P
       1. femmepk -> shibil -> 12345
    4. psql
    5. grant all privileges on database femmepk to shibil;
    6. \q
    7. exit
11. Create super user
    1. python manage.py createsuperuser
       1. shibil -> shibil789
12. Migration comments
    1. python manage.py makemigrations
    2. python manage.py migrate
    3. python manage.py loaddata initial\_data user\_groups permissions notification
13. Delete migration
    1. find . -path "\*/migrations/\*.py" -not -name "\_\_init\_\_.py" -delete
    2. find . -path "\*/migrations/\*.pyc" -delete
14. Read and write from file
    1. pip freeze > r.txt
    2. pip install -r r.txt
    3. python manage.py loaddata initial\_data user\_groups permissions notification

Django documentation notes

1. Models objects return functions
   1. all()
   2. filter()
      1. filter value ( instance= , )
   3. exclude()
      1. To avoid some fields
   4. get()
      1. for one value
2. Models objects create functions
   1. [create](#74m6yrb1me5e)()
3. Forms for single line
   1. for field in form:

field.id\_for\_label || field.label || field

endfor

* 1. form.as\_p

1. Form validation views.py(function)
2. if request.method=="POST":

name = request.POST.get('name')

email = request.POST.get('email')

FormClassName.objects.create(

name = name,

email = email,

)

return HttpResponse('Form Submitted')

else:

return HttpResponse('Invalid Request')

Django Pagination

1. pip install django-el-pagination
2. Register at INSTALLED\_APP in setting -
   1. ‘el\_pagination ’
3. In template
   1. {% load el\_pagination\_tags %}
   2. {% paginate 21 instances %}
   3. set loader temp

Auto Complete

1. From App
   1. urls.py
      1. from whichapp.views import NameAutocomplete
      2. urlpatterns = [ url(r'^anyname-autocomplete/$',NameAutocomplete.as\_view(),name='urlname\_autocomplete',),
   2. views.py
      1. from dal import autocomplete
      2. class NameAutocomplete(autocomplete.Select2QuerySetView):

def get\_queryset(self):

if not self.request.user.is\_authenticated():

return whichapp.objects.none()

items = whichapp.objects.filter(is\_deleted=False)

if self.q:

query = self.q

items = items.filter(Q(name\_\_icontains=query) | Q(phone\_\_icontains=query) | Q(email\_\_icontains=query) | Q(address\_\_icontains=query)

)

return items

1. To App
   1. forms.py
      1. 'fieldname' : autocomplete.ModelSelect2(url='mainappulrname:urlname\_autocomplete ',attrs={'data-placeholder': 'Product','data-minimum-input-length': 0},),
2. {{ formname.media }}

User Authentication

1. pip install django-registration-redux
2. In main setting
   1. register app - ‘registration’,
   2. LOGIN\_URL = '/seturl/url/'

LOGOUT\_URL = '/seturl/url/'

LOGIN\_REDIRECT\_URL = '/'

1. main url.py
   1. url(r'^app/accounts/', include('registration.backends.default.urls')),