Tutorial Django sistema Algorisoft con postgresql

1. Se crea una carpeta
2. Se crea el ambiente 🡪 Python -m venv venv
3. Se activa el ambiente 🡪 activate
4. Se instala freeze en la carpeta scripts 🡪 pip install freeze
5. Se instala django en la carpeta scripts 🡪 pip install django
6. En la raíz se crea el projecto 🡪 django-admin startproject app
7. Se ejecuta el Project para crear la bd 🡪 Python manage.py runserver
8. Se crea un archivo txt para que nos guarde todas las librerías en ese archivo:

pip freeze > requerimientos.txt

1. En config se crea un archivo llamada db.py para guardar configuraciones de las bd

import os  
  
BASE\_DIR= os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))  
  
SQLITE={  
 'default':{  
 'ENGINE':'django.db.backends.sqlite3',  
 'NAME': os.path.join(BASE\_DIR, 'db/sqlite/db.sqlite3'),  
 }  
}  
  
POSTGRESQL={  
 'default':{  
 'ENGINE':'django.db.backends.postgresql\_psycopg2',  
 'NAME': 'db\_horus',  
 'USER': 'postgres',  
 'PASSWORD': '12345',  
 'HOST': 'localhost',  
 'PORT': '5432'  
 }  
}

1. Se instala el driver para utilizar postgres

pip install psycopg2

1. Se crea la base de datos en postgres db\_horus
2. Luego vamos al archivo settings.py y se reemplaza lo siguiente:

El from pathlib… se quita :

from pathlib import Path  
  
# Build paths inside the project like this: BASE\_DIR / 'subdir'.  
BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

y se reemplaza por esto:

import os

import app.db as db

BASE\_DIR= os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))

1. En la terminal se realizan las migraciones

Python manage.py makemigrations

Python manage.py migrate

1. Actualizamos el archivo de requerimientos 🡪

pip freeze > requirements/requirements.txt

1. Se crea una carpeta llamada “core” para almacenar cada una de las apps que se van a crear
2. Creo la primera app llamada 🡪 django-admin startapp erp
3. Se empiezan a crear los modelos en el archivo models.py

from django.db import models

from datetime import datetime

class Employee(*models*.*Model*):

names=models.CharField(*max\_length*=150, *verbose\_name*='Nombres')

cedula=models.CharField(*max\_length*=10, *unique*=True, *verbose\_name*='Cedula')

date\_joined=models.DateField(*default*=datetime.now, *verbose\_name*='Fecha de Registro') #fecha en la que se creó el registro

date\_created=models.DateTimeField(*auto\_now*=True) #pone la fecha actual

date\_updated=models.DateTimeField(*auto\_now\_add*=True) #actualiza la fecha

age=models.PositiveIntegerField(*default*=0) #tipo de datos positivos

salary=models.DecimalField(*default*=0.00, *max\_digits*=9, *decimal\_places*=2)

status=models.BooleanField(*default*=True)

avatar=models.ImageField(*upload\_to*='avatar/%y/%m/%D', *null*=True, *blank*=True)

cv=models.FileField(*upload\_to*='avatar/%y/%m/%D', *null*=True, *blank*=True)

def \_\_str\_\_(*self*):

return *self*.names

class Meta:

        verbose\_name='Empleado'

        verbose\_name\_plural='Empleados'

        db\_table='empleado'

        ordering=['id']

1. En el archivo settings.py se agregan las apps

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    #Apps

    'core.erp'

]

1. Para trabajar con imágenes se debe instalar la librería pillow en scripts

pip install pillow

1. Se actualiza el requirements.txt

pip freeze > requirements/requirements.txt

1. Se realizan las migraciones

python manage.py makemigrations

python manage.py migrate

1. Se crea el control de versiones en gitlab. Se abre el gitbash de git ya instalado en el pc y se crea un ssh para vincuarlo con gitlab.

* Se crea el ssh

$ ssh-keygen -t rsa -b 4096

\*se copia este comando

$ cat ~/.ssh/id\_rsa.pub

* Se Copia la clave ssh
* Vamos a github y buscamos settings. En settings ponemos ‘ssh keys’ y vinculamos la clave ssh que copiamos anteriormente.
* Luego vamos a crear un nuevo proyecto.
* Damos en Clone. Copiamos el el link de ssh.
* Vamos a la ruta donde tenemos el proyecto y clonamos
* Git status, git add -f app/, git commit -m "add folder app"