برای ساخت اسلایدر باید اپی برای آن ساخت و مدل Slide را پیاده سازی کرد و درنهایت برای ساخت هر اسلاید ازش استفاده میکنیم

Model:

```python
import os
import time
from django.db import models

def get_name_extention(filepath):...

def upload_image_path(instance, filepath):...

class Slide(models.Model):
    title = models.CharField(max_length=70,verbose_name='عنوان')
    titleWhite = models.CharField(max_length=70,verbose_name='عنوان دوم',blank=True)
    desc = models.TextField(max_length=250,verbose_name="توضیحات")
    link = models.URLField(max_length=150,verbose_name='لینک')
    image = models.ImageField(upload_to=upload_image_path,verbose_name='تصویر')
    active = models.BooleanField(default=True, verbose_name='فعال')

    class Meta:
        verbose_name='اسلایدر'
        verbose_name_plural='اسلایدر'

    def __str__(self):
        return self.title
```

View: (in homepage or index page)

```python
from django.shortcuts import render
from eshop_sliders.models import *

def header(request, *args, **kwargs):
    contect={
        'data':'در ایام کرونا با خرید آنلاین از شیوع بیشتر این بیماری جلوگیری کنید',
    }
    return render(request, 'shared/Header.html', contect)

def footer(request, *args, **kwargs):
    context={
        'about_us':'تمامی حقوق این وب سایت متعلق به رضا رضایی میباشد',
    }
    return render(request, 'shared/Footer.html', context)

def home(request):
    slider=Slide.objects.all()      ⬅
    context={
        'slider':slider,        ⬅
    }
    return render(request, 'homepage.html', context)
```

Template: (depended to html template kind)

```html
{% extends 'shared/_MainLayout.html' %}
{% load static %}
{% block content %}
    <section id="slider"><!--slider-->
    <div class="container">
        <div class="row">
            <div class="col-sm-12">
                <div id="slider-carousel" class="carousel slide" data-ride="carousel">
                    <ol class="carousel-indicators">
                        {% for slide in slider %}
                            <li data-target="#slider-carousel" data-slide-to="{{ forloop.counter0 }}" class="{{ forloop.first|yesno:"active," }}"></li>
                        {% endfor %}
                    </ol>
                    <div class="carousel-inner">
                        {% for slide in slider %}
                            <div class="item {{ forloop.first|yesno:"active," }}">
                                <div class="col-sm-6">
                                    <h1> {{ slide.title }} <span>{{ slide.titleWhite }}</span></h1>
                                    <p style="...">{{ slide.desc|truncatechars:200}}</p>
                                    <a class="btn btn-default get" href="{{ slide.link }}">بیشتر</a>
                                </div>
                                <div class="col-sm-6">
                                    <img style="..." src="{{slide.image.url}}" class="girl img-responsive" alt="" />

                                </div>
                            </div>
                        {% endfor %}
                    </div>

                    <a href="#slider-carousel" class="left control-carousel hidden-xs" data-slide="prev">
                        <i class="fa fa-angle-left"></i>
                    </a>
                    <a href="#slider-carousel" class="right control-carousel hidden-xs" data-slide="next">
                        <i class="fa fa-angle-right"></i>
                    </a>
                </div>

            </div>
        </div>
    </div>
</section><!--/slider-->
```

section#slider › div.container › div.row › div.col-sm-12 › div#slider-carousel.carousel.slide › ol.carousel-indicators

* If each slide has "active" class is shown it, else not displayed

- so should first slide has "active" class and other if turned page be first and "active".

+ for make this structure use jinja filters:

   1. forloop.first -> give first object
   2. …|yesno:"active," -> give Boolean result of check condition And put answer in double quote depended result and separate by comma

* the list of slide's receive in view.py from backend.