

```

# mandelbrot/views.py

from django.shortcuts      import render
from django.views.generic import TemplateView
from django.shortcuts      import render_to_response
from django.template       import RequestContext

from reportlab.pdfgen      import canvas
from django.http           import HttpResponse
from django.http           import HttpResponseRedirect

from django.contrib.auth.decorators import login_required
from django.contrib.auth.mixins     import LoginRequiredMixin
from django.contrib.auth.models     import User

# Create your views here.

class HomePageView(TemplateView):
    def get(self, request, **kwargs):
        return render(request, 'index.html', context=None)

class LoginPageView(TemplateView):
    print("written from LoginPageView in views.py")
    template_name = "login.html"

class LogoutPageView(TemplateView):
    print("written from LogoutPageView in views.py")
    template_name = "logout.html"

#login required to see CodePageView
class CodePageView(LoginRequiredMixin, TemplateView):
    print("written from CodePageView in views.py")
    template_name = "code.html"

#login required to see ColorPageView
#class ColorPageView(LoginRequiredMixin, TemplateView):
#    print("written from ColorPageView in views.py")
#    template_name = "color.html"

#def ProfilePageView(request):
#    url = request.user.profile.url

#class PdfPageView(TemplateView):
#    print("written from PdfPageView in views.py")
#    template_name = "pdf.html"

@login_required
def ColorPageView(request):
    user = request.user
    print("change color for username: ",user.username)

    color = request.POST['changeColor']
    print("change to: ", color)
    print("current user.profile.color", user.profile.color)

    user.profile.color = color
    user.save()
    print("new user.profile.color", user.profile.color)

    return HttpResponseRedirect('/')

```

```
@login_required
def PdfPageView(request):

    # Create the HttpResponse object with the appropriate PDF headers.
    response = HttpResponse(content_type='application/pdf')
    response['Content-Disposition'] = 'attachment; filename="mandelbrot.pdf"'

    # Create the PDF object, using the response object as its "file."
    p = canvas.Canvas(response)

    # Draw things on the PDF. Here's where the PDF generation happens.
    # See the ReportLab documentation for the full list of functionality.
    p.drawString(100, 50, "Hello world.")

    # Close the PDF object cleanly, and we're done.
    p.showPage()
    p.save()

    return response
```