Login and registration

Models.py

# -\*- coding: utf-8 -\*-

from \_\_future\_\_ import unicode\_literals

import re

import bcrypt

from django.db import models

EMAIL\_REGEX = re.compile(r'^[a-zA-Z0-9\.\+\_-]+@[a-zA-Z0-9\.\_-]+\.[a-zA-Z]\*$')

NAME\_REGEX = re.compile(r'^[A-Za-z]\w+$')

class UserManager(models.Manager):

def validate\_login(self, post\_data):

errors = []

# check DB for post\_data['email']

if len(self.filter(email=post\_data['email'])) > 0:

# check this user's password

user = self.filter(email=post\_data['email'])[0]

if not bcrypt.checkpw(post\_data['password'].encode(), user.password.encode()):

errors.append('email/password incorrect')

else:

errors.append('email/password incorrect')

if errors:

return errors

return user

def validate\_registration(self, post\_data):

errors = []

# check length of name fields

if len(post\_data['first\_name']) < 2 or len(post\_data['last\_name']) < 2:

errors.append("name fields must be at least 3 characters")

# check length of name password

if len(post\_data['password']) < 8:

errors.append("password must be at least 8 characters")

# check name fields for letter characters

if not re.match(NAME\_REGEX, post\_data['first\_name']) or not re.match(NAME\_REGEX, post\_data['last\_name']):

errors.append('name fields must be letter characters only')

# check emailness of email

if not re.match(EMAIL\_REGEX, post\_data['email']):

errors.append("invalid email")

# check uniqueness of email

if len(User.objects.filter(email=post\_data['email'])) > 0:

errors.append("email already in use")

# check password == password\_confirm

if post\_data['password'] != post\_data['password\_confirm']:

errors.append("passwords do not match")

if not errors:

# make our new user

# hash password

hashed = bcrypt.hashpw((post\_data['password'].encode()), bcrypt.gensalt(5))

new\_user = self.create(

first\_name=post\_data['first\_name'],

last\_name=post\_data['last\_name'],

email=post\_data['email'],

password=hashed

)

return new\_user

return errors

class User(models.Model):

first\_name = models.CharField(max\_length=100)

last\_name = models.CharField(max\_length=100)

email = models.EmailField(unique=True)

password = models.CharField(max\_length=255)

objects = UserManager()

def \_\_str\_\_(self):

return self.email

urls.py

from django.conf.urls import url

from . import views

urlpatterns = [

url(r'^$', views.index),

url(r'^login$', views.login),

url(r'^register$', views.register),

url(r'^success$', views.success),

]

Views.py

# -\*- coding: utf-8 -\*-

from \_\_future\_\_ import unicode\_literals

from .models import User

from django.shortcuts import render, redirect

from django.contrib import messages

# Create your views here.

def index(request):

return render(request, 'login/index.html')

def register(request):

result = User.objects.validate\_registration(request.POST)

if type(result) == list:

for err in result:

messages.error(request, err)

return redirect('/')

request.session['user\_id'] = result.id

messages.success(request, "Successfully registered!")

return redirect('/success')

def login(request):

result = User.objects.validate\_login(request.POST)

if type(result) == list:

for err in result:

messages.error(request, err)

return redirect('/')

request.session['user\_id'] = result.id

messages.success(request, "Successfully logged in!")

return redirect('/success')

def success(request):

try:

request.session['user\_id']

except KeyError:

return redirect('/')

context = {

'user': User.objects.get(id=request.session['user\_id'])

}

return render(request, 'login/success.html', context)