**Forms.py**

**from** django.shortcuts **import** render  
**from** django.template **import** RequestContext  
  
*# Create your views here.***def** test\_flowcell(request):  
 c = RequestContext(request, {**'other\_context'**:**'details here'**})  
 **if** request.method == **'POST'**: *# If the form has been submitted...* form = ImportExcelForm(request.POST, request.FILES) *# A form bound to the POST data* **if** form.is\_valid(): *# All validation rules pass* excel\_parser= ExcelParser()  
 success, log = excel\_parser.read\_excel(request.FILES[**'file'**] )  
 **if** success:  
 **return** redirect(reverse(**'admin:index'**) + **"pages/flowcell\_good/"**) *## redirects to aliquot page ordered by the most recent* **else**:  
 errors = **'\* Problem with flowcell \* <br><br>log details below:<br>'** + **"<br>"**.join(log)  
 c[**'errors'**] = mark\_safe(errors)  
 **else**:  
 c[**'errors'**] = form.errors  
 **else**:  
 form = ImportExcelForm() *# An unbound form* c[**'form'**] = form  
 **return** render\_to\_response(**'sequencing/file\_upload.html'**)

models.py

**from** django.db **import** models  
**import** random  
**import** os  
**from** django.db **import** models  
**from** django.db.models **import** Q  
**from** django.db.models.signals **import** pre\_save, post\_save  
**from** django.urls **import** reverse  
**import** pandas **as** pd  
  
*# def get\_filename\_ext(filepath):  
# base\_name = os.path.basename(filepath)  
# name, ext = os.path.splitext(base\_name)  
# return name, ext  
#  
# def upload\_file\_path(instance, filename):  
# new\_filename = random.randint(1, 3910209312)  
# name, ext = get\_filename\_ext(filename)  
# final\_filename = '{new\_filename}{ext}'.format(new\_filename=new\_filename, ext=ext)  
#  
# return "upload/{new\_filename}/ {final\_filename}".format(new\_filename=new\_filename, final\_filename=final\_filename)***class** Upload(models.Model):  
 *#file = models.FileField(upload\_to='upload\_file\_path', null=True, blank=True)* file = models.FileField(default=**True**)  
 print(file.name)  
 **def** \_\_str\_\_(self):  
 **return** os.path.basename(self.file.name)  
  
 **def** save(self, \*args, \*\*kwargs):  
 super(Upload, self).save(\*args, \*\*kwargs)  
 filename = self.file.url  
 **def** vin(self):  
 dict = pd.read\_csv(**"{file}"**)