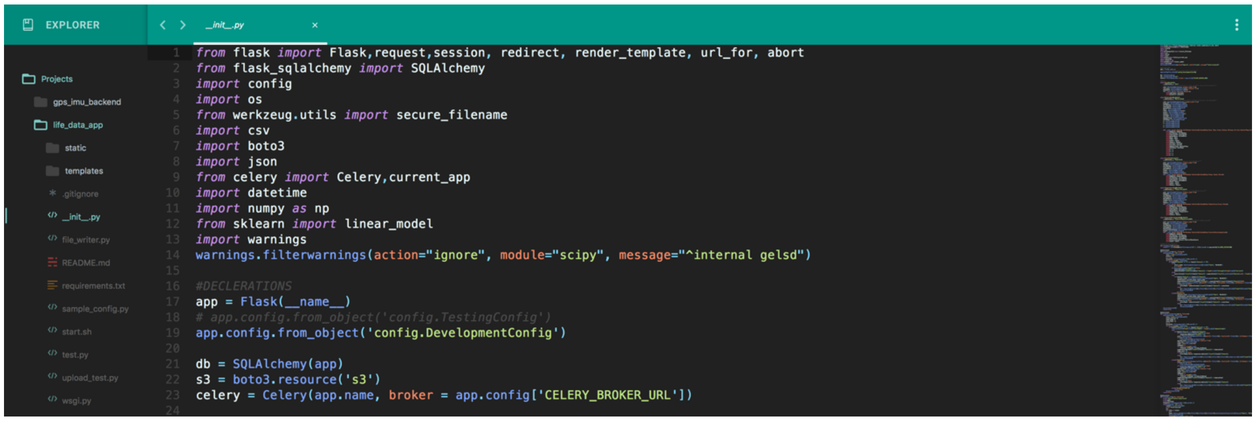
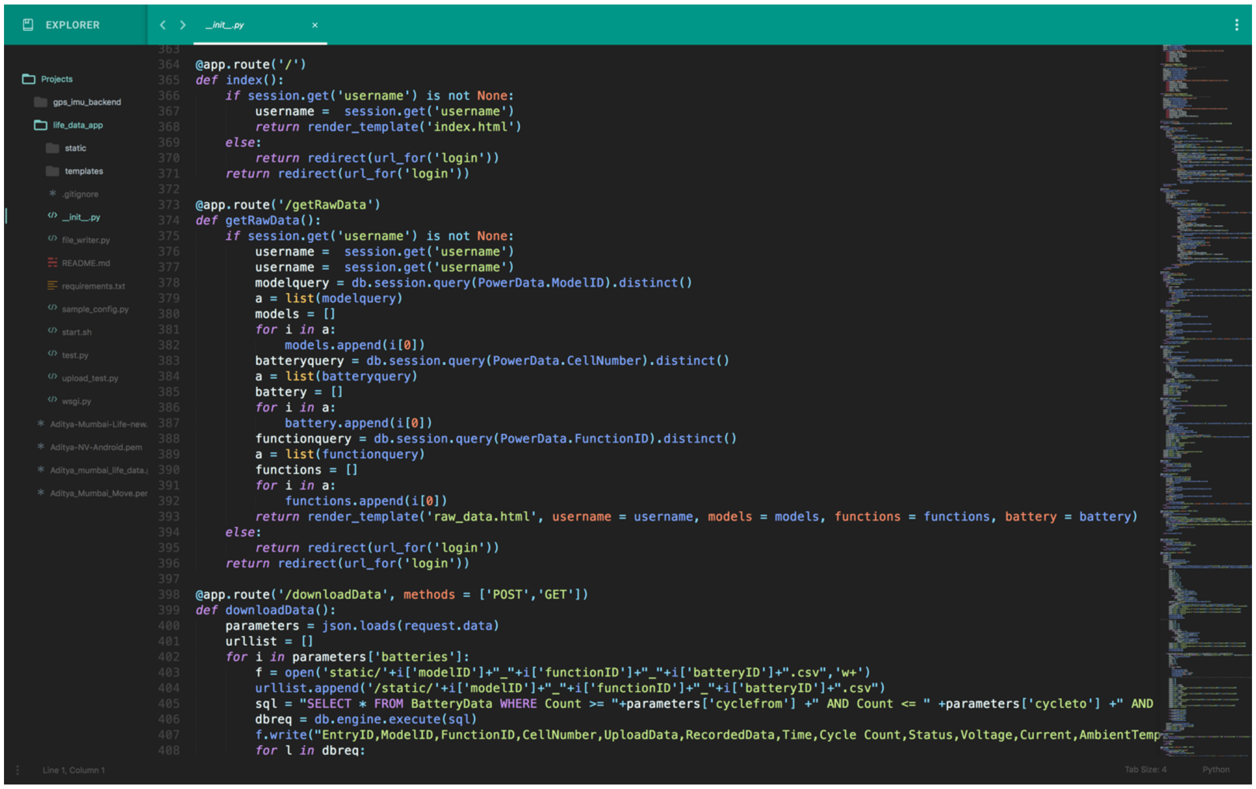


```python
from flask import Flask,request,session, redirect, render_template, url_for, abort
from flask_sqlalchemy import SQLAlchemy
import config
import os
from werkzeug.utils import secure_filename
import csv
import boto3
import json
from celery import Celery,current_app
import datetime
import numpy as np
from sklearn import linear_model
import warnings
warnings.filterwarnings(action="ignore", module="scipy", message="^internal gelsd")

#DECLERATIONS
app = Flask(__name__)
# app.config.from_object('config.TestingConfig')
app.config.from_object('config.DevelopmentConfig')

db = SQLAlchemy(app)
s3 = boto3.resource('s3')
celery = Celery(app.name, broker = app.config['CELERY_BROKER_URL'])
```

Fig7. Modules Used



```python

@app.route('/')
def index():
    if session.get('username') is not None:
        username =  session.get('username')
        return render_template('index.html')
    else:
        return redirect(url_for('login'))
    return redirect(url_for('login'))

@app.route('/getRawData')
def getRawData():
    if session.get('username') is not None:
        username =  session.get('username')
        username =  session.get('username')
        modelquery = db.session.query(PowerData.ModelID).distinct()
        a = list(modelquery)
        models = []
        for i in a:
            models.append(i[0])
        batteryquery = db.session.query(PowerData.CellNumber).distinct()
        a = list(batteryquery)
        battery = []
        for i in a:
            battery.append(i[0])
        functionquery = db.session.query(PowerData.FunctionID).distinct()
        a = list(functionquery)
        functions = []
        for i in a:
            functions.append(i[0])
        return render_template('raw_data.html', username = username, models = models, functions = functions, battery = battery)
    else:
        return redirect(url_for('login'))
    return redirect(url_for('login'))

@app.route('/downloadData', methods = ['POST','GET'])
def downloadData():
    parameters = json.loads(request.data)
    urllist = []
    for i in parameters['batteries']:
        f = open('static/'+i['modelID']+"_"+i['functionID']+"_"+i['batteryID']+".csv",'w+')
        urllist.append('/static/'+i['modelID']+"_"+i['functionID']+"_"+i['batteryID']+".csv")
        sql = "SELECT * FROM BatteryData WHERE Count >= "+parameters['cyclefrom'] +" AND Count <= " +parameters['cycleto'] +" AND
        dbreq = db.engine.execute(sql)
        f.write("EntryID,ModelID,FunctionID,CellNumber,UploadData,RecordedData,Time,Cycle Count,Status,Voltage,Current,AmbientTemp
        for l in dbreq:
```

Fig8. Application Routes