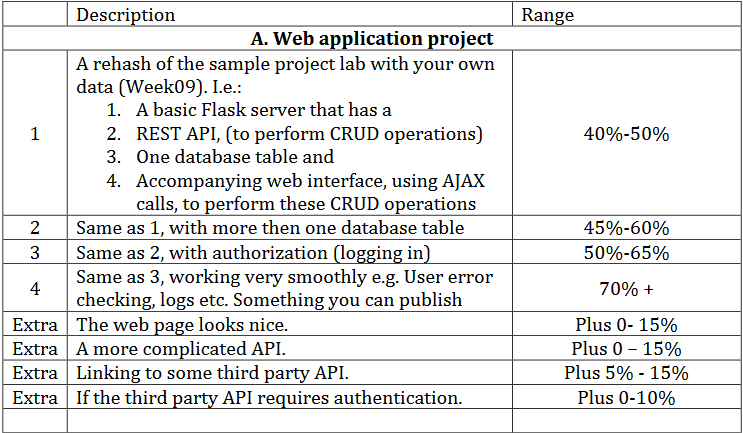
**Etain’s final submission for the ‘big project’ as part of module 52957 - DATA-REPRESENTATION.**

**Lecturer: Andrew Beatty**

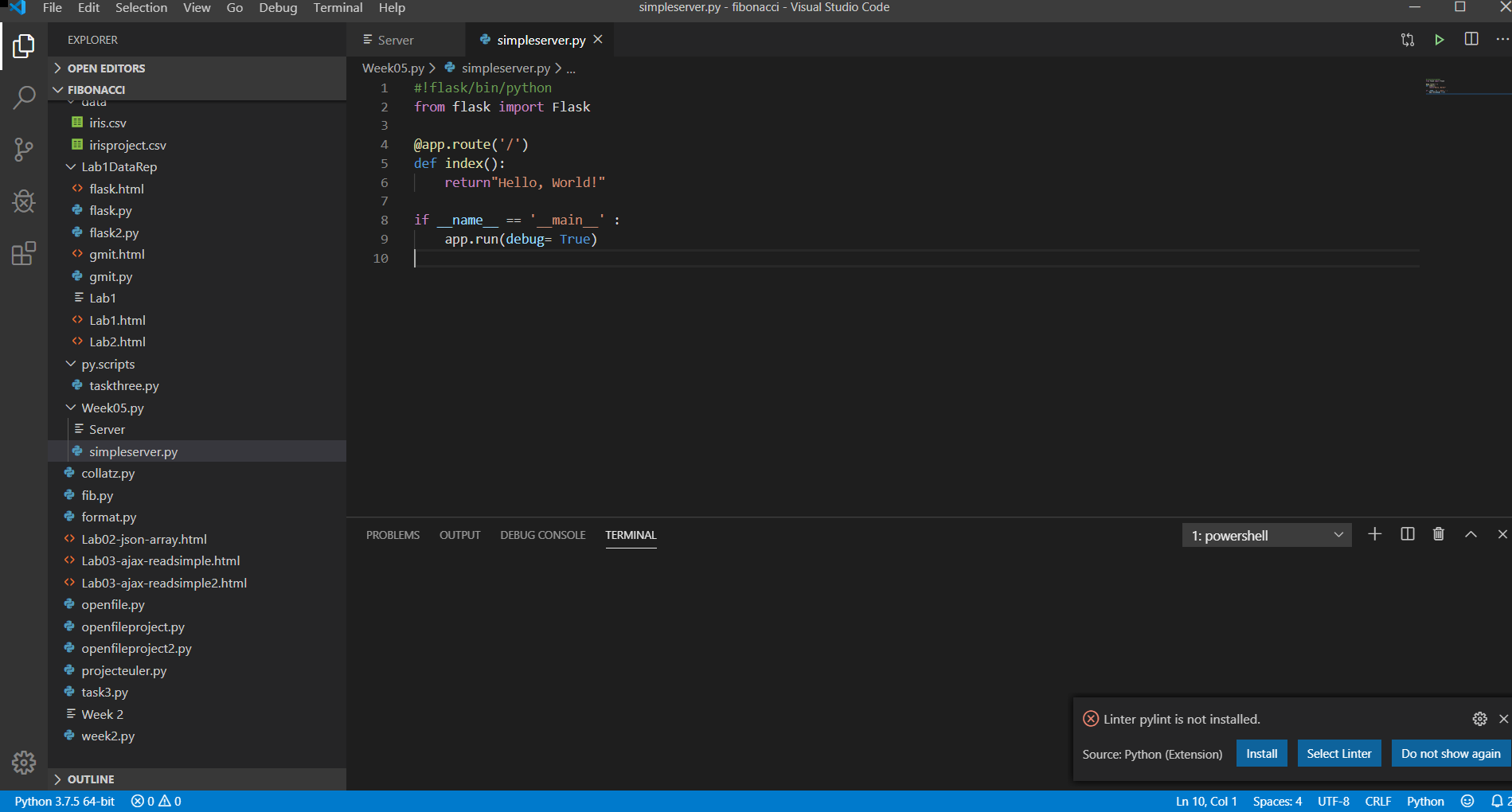
**Student: Etain Upton G00318876**

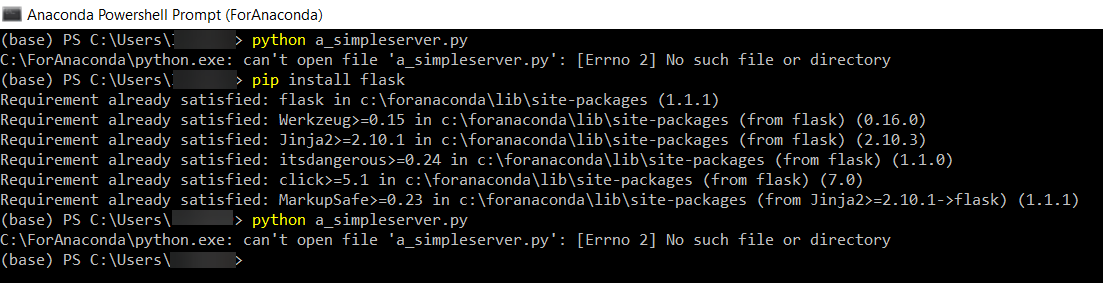
**I chose to complete Option (A) for this assessment – a Web Application based on;**



I found this module challenging. I successfully managed to complete Lab 01 but struggled with Lab 02 onwards.

I ran into issues following the *Lab 05.2 Flask* instructions;





As shown above, I continued to get the error ‘C:\ForAnaconda\python.exe: can't open file 'a\_simpleserver.py': [Errno 2] No such file or directory’.

I struggled to progress past this.

I have included my code below for review.

**An app server that has a RESTful interface to provide CRUD operations for one database table;**

I followed the steps completed by Andrew Beatty in his lecture *‘DR8.2a REST for project’* to write the following code;

from flask import Flask, jsonify, request, abort

app = Flask(\_\_name\_\_, static\_url\_path='', static\_folder='.')

dogs=[

    { "id":123, "Name":"Fluffy", "Owner":"John Doe", "Value": 10000},

    { "id":456, "Name":"Bob", "Owner":"Jane Doe", "Value": 20000},

    { "id":789, "Name":"Rex", "Owner":"Jim Doe", "Value": 30000}

]

nextId=101

#app = Flask(\_\_name\_\_)

#@app.route('/')

#def index():

#    return "Hello, World!"

@app.route('/dogs')

def getAll():

    return jsonify(dogs)

@app.route('/dogs/<int:id>')

def findById(id):

    foundDogs = list(filter(lambda b: b['id'] == id, dogs))

    if len(foundDogs) == 0:

        return jsonify ({}) , 204

    return jsonify(foundDogs [0])

@app.route('/dogs', methods=['POST]')

def create():

    global nextId

    if not request.json:

        abort(400)

    #other checking

    dog = {

        "id": nextId,

        "Name": request.json['Name'],

        "Owner": request.json['Owner'],

        "Value": request.json['Value']

    }

    nextId += 1

    dogs.append(dog)

    return jsonify(dog)

@app.route('/dogs/<int:id>', methods=['PUT'])

def update(id):

    foundDogs = list(filter(lambda t: t['id']== id, dogs))

    if (len(foundDogs) == 0):

        abort(404)

    foundDog = foundDogs[0]

    if not request.json:

        abort(400)

    reqJson = request.json

    if 'Value' in reqJson and type(reqJson['Value']) is not int:

        abort(400)

    if 'Name' in reqJson:

        foundDog['Name'] = reqJson['Name']

    if 'Owner' in reqJson:

        foundDog['Owner'] = reqJson['Owner']

    if 'Value' in reqJson:

        foundDog['Value'] = reqJson['Value']

    return jsonify(foundDog)

    return "in update for id "+str(id)

@app.route('/dogs/<int:id>', methods=['DELETE'])

def delete(id):

    foundDogs = list(filter(lambda t: t['id']== id, dogs))

    if (len(foundDogs) == 0):

        abort(404)

    dogs.remove(foundDogs[0])

    return jsonify({"done":True})

if \_\_name\_\_ == '\_\_main\_\_' :

    app.run(debug= True)