Name:

Last 4 digits ID:

Computer IP address:

CSE 6331, Cloud Computing

Quiz Q1, Summer 2018 (c) DL, UTA, 2018

C&P means cut and paste only those relevant lines from your program(s) into this quiz.

1. I understand that I am on my honor during this quiz, I will not collaborate, use

non-allowed sources, and I will not discuss or share this quiz with anyone for the next

4 hours.

You MAY: use Google to search, use previous source code,

YOU MAY NOT use:

Email, Facebook, Hangouts, IM, chats, Skype or ANY other human connection.

This is a timed test. Late tests will have points deducted for being late.

Very late tests will not be graded.

When you are complete, with any part, please raise your hand, so we may visually inspect that part.

The second part of the test, you should electronically submit, you will need to copy and paste

only those lines of code to implement that part of the test, usually a few (two to eight) lines of code.

Place it immediately after the question.

Submit this Quiz (renamed) with code cut and pasted, ONLY text. DO NOT submit zips, binaries, libraries,

or anything other than text.

When any parts(questions) are complete complete, submit this test, you may make multiple submissions.

If you have computer, or other, problems, please raise your hand immediately.

If you understand and agree, please initial here:

**SA**

2. Get files from this same folder.

3. Name your program with your name and last digits of your ID.

4. Using the cloud services provider, all functionality possible should be implemented on that provider.

(Of course, displaying web pages through a browser and the user interface is "local")

Show and submit code:

5. Show a web page and interface (which resides on the cloud provider) with your name and

student ID in large font at the top of every web page dispayed (for this quiz)

For the first page also show a photo (picture) of you, if you do not have one you may use

the photo "me.jpg".

<h1>Siddharth Agarwal</h1>

<h1>7570</h1>

<img src="/static/images/me.jpg" alt="">

6. Upload all data files to cloud service, then show sizes (on cloud) of each on your screen.

@app.route('/api/createcsv', methods=['GET', 'POST'])

def csv():

connection = sqlite3.connect("people.db")

cursor = connection.cursor()

if request.method == 'POST':

if 'csv' not in request.files:

result = ["no file found, please upload again"]

return jsonify(results=result)

else:

table\_name = "student"

file = request.files['csv']

df = pandas.read\_csv(file.stream)

df.to\_sql(table\_name, connection, if\_exists='append', index=False)

return jsonify(results=df.to\_json())

if request.method == 'GET':

query = "SELECT \* FROM student"

# query2 = "SELECT name,picture FROM student where grade=98"

cursor.execute(query)

results = cursor.fetchall()

connection.close()

# print(len(results))

return render\_template('displaydbdata.html', students = results, size=len(results))

7. Allow a user, through a web form, to give a course number and show the names of the instructors.

@app.route('/search/instructor', methods=['GET', 'POST'])

def searchInstructor():

if request.method == 'GET':

return render\_template('searchinstructor.html')

if request.method == 'POST' :

if 'course' not in request.form:

flash('No course given')

return redirect(request.url)

# result = ["no photo found, please upload again"]

# return jsonify(results=result)

else:

connection = sqlite3.connect("people.db")

cursor = connection.cursor()

query = "select instructor from student where Course = {0} distinct".format(int(request.form['course']))

# query2 = "SELECT name,picture FROM student where grade=98"

cursor.execute(query)

results = cursor.fetchall()

connection.close()

return jsonify(results=results)

8. Allow a user to give Instructor's name (web form), show all courses, sections, and rooms that person

is assigned to.

@app.route('/search/detail', methods=['GET', 'POST'])

def searchDetail():

if request.method == 'GET':

return render\_template('detail.html')

if request.method == 'POST' :

if 'instructor' not in request.form:

flash('No course given')

return redirect(request.url)

# result = ["no photo found, please upload again"]

# return jsonify(results=result)

else:

connection = sqlite3.connect("people.db")

cursor = connection.cursor()

instructor = request.form['instructor']

# print (instructor)

# query = "select course,section,room from student where instructor='Kashefi'"

query = "select course,section,room from student where instructor="+"'"+request.form['instructor'] +"'"

# query2 = "SELECT name,picture FROM student where grade=98"

cursor.execute(query)

results = cursor.fetchall()

connection.close()

return render\_template('detail.html',results=results)

# return jsonify(results=results)

9. Allow a user to give range of course numbers (in a web form), list total number of courses in that range and

instructors names (just list each instructor once).

@app.route('/search/range', methods=['GET', 'POST'])

def searchRange():

if request.method == 'GET':

return render\_template('range.html')

if request.method == 'POST' :

if 'instructor' not in request.form:

flash('No range given')

return redirect(request.url)

# result = ["no photo found, please upload again"]

# return jsonify(results=result)

else:

connection = sqlite3.connect("people.db")

cursor = connection.cursor()

lower = request.form['lower']

upper = request.form['upper']

# print (instructor)

# query = "select course,section,room from student where instructor='Kashefi'"

query = "select \* from student where course >"+"'"+lower +"'" + "and course < " +"'"+upper +"' distinct"

# query2 = "SELECT name,picture FROM student where grade=98"

cursor.execute(query)

results = cursor.fetchall()

connection.close()

return render\_template('detail.html',results=results)

# return jsonify(results=results)

10. Allow a user to select a course number and section name and then change instructors name and room.

11. Show GTA parts 5,6, 7, 8, 9, 10

12. When complete, return (send) this quiz

If you finish early, send this immediately, otherwise send between

the end of class and no more than 1 minute after that.