Introduction to Computer Science Syllabus

Lab 24: Twitter

May 1, 2018

Main Event

1. Display tweets

Display all of the tweets currently in the database as an HTML table. How you order the columns is up to you. You should display all information related to a tweet.

2. Tweet!

Above your table, display a form consisting of three elements:

1. an input box where a user will enter their user name;
2. an input box where a user will enter their tweet; and
3. a submit button.

The form should POST the information back to the page.

```
<form method="POST" action="/">
```

3. Tweet filters

Beneath your tweet form, but still above the tweet display table, build another form. This form will allow you to filter tweets by a given user, and to control the number of tweets displayed. The form should contain the the following three elements:

1. A selection menu (HTML option) containing all unique Twitter users in the database, in ascending order. The first user listed should be a special user that is not in the database. Selection of that user should signify that no user filtration should take place.

2. A text box in which the user can enter the number of tweets they would like to display.

3. A submit button.

Unlike the previous form, this form should be submitted using as a GET request.

```
<form method="GET" action="/">
```

4. Like!

Retool the ID column to allow the user to “like” that particular tweet.

1. Remove the table heading for ID. You can replace the heading with “Like,” or just leave it empty.

2. Instead of displaying the tweets ID, display the word “like.” That word should actually be a link back to the page itself defining a query string that informs the page to increment the number of likes for that tweet ID. Thus, although this column no longer displays the ID, it still depends on it.

   If you are confused, recall the counter example we did in class in which the anchor tag was used to increment a counter.

Solution:

```
import flask

import mysql.connector as db

app = flask.Flask(__name__)

def tw_form():
    return """
<form method="POST" action="/">
<table width="100%"><tr>
<td>User: <input type="text" name="author"></td>
<td>Message: <textarea name="tweet"></textarea></td>
<td><input type="submit" value="Tweet!">
</tr></table>
</form>
"""
```

```
def tw_filters(cursor, args):
    cursor.execute('SELECT DISTINCT(user) AS u FROM tweet ORDER BY u')

    form = [
        '<form method="GET" action="/">',
        '<table width="100%"><tr>',
        '<td>Show only tweets from <select name="user">',
    ]
    form.append('<option value=""></option>')

    for row in cursor:
        if 'user' in args and args['user'] == row['u']:
            selected = 'selected'
        else:
            selected = ''

        form.append('<option value="{0}" {1}>{0}</option>'.format(row[
                                                                     select
    form.append('</select></td>')

    form.extend([
        '<td>Limit retrieved tweets to: <input type="text" name="limit
        '<td><input type="submit" value="Filter"></td>',
        '</tr></table>',
        '</form>',
    ])

    return '\n'.join(form)

def tw_show(cursor, args):
    attributes = [ 'id', 'tstamp', 'user', 'message', 'likes' ]
    sql = [
        'SELECT {0}'.format(','.join(attributes)),
        'FROM tweet',
    ]
    if 'user' in args and args['user']:
        sql.append("WHERE user = '{0}'".format(args['user']))
    sql.append('ORDER BY tstamp DESC')
    if 'limit' in args and args['limit'].isdigit():
        sql.append('LIMIT {0}'.format(args['limit']))
    sql = ' '.join(sql)
    print(sql)
    cursor.execute(sql)

    table = [
        '<table width="100%" cellpadding="5">'
        '<tr bgcolor="#c0c0c0">',
        '<td>&nbsp;</td>',
        '<td>' + '</td><td>'.join(attributes[1:]) + '</td>',
        '</tr>',
    ]

    for row in cursor:
        table.append('<tr>')
        for i in attributes:
            wrap = ''
            if i == 'id':
                value = '<a href="/?like={0}">like</a>'.format(row['ic
            elif i == 'tstamp':
                value = row['tstamp'].strftime('%c')
                value = value.replace(' ', '&nbsp;')
                wrap = 'nowrap="nowrap"'
            else:
                value = row[i]
            table.append('<td {1}>{0}</td>'.format(value, wrap))
        table.append('</tr>')

    return '\n'.join(table)

@app.route('/', methods=['GET', 'POST'])
def html():
    get = flask.request.args
    post = flask.request.form

    connection = db.connect(host='10.224.45.113',
                            user='cs101',
                            db='twitter',
                            autocommit=True)
    cursor = connection.cursor(dictionary=True)
    # insert
    if 'tweet' in post:
        values = "('{0}', '{1}')".format(post['author'], post['tweet']
        sql = 'INSERT INTO tweet (user, message) VALUES ' + values
        cursor.execute(sql)
    # update
    if 'like' in get and get['like']:
        sql = [
            'UPDATE tweet SET likes = likes + 1',
            'WHERE id = {0}'.format(get['like'])
        ]
        cursor.execute(' '.join(sql))

    table = [
        '<table width="100%">',
        '<tr>{0}</tr>'.format(tw_form()),
        '<tr><hr></tr>',
        '<tr>{0}</tr>'.format(tw_filters(cursor, get)),
        '<tr><hr></tr>',
        '<tr>{0}</tr>'.format(tw_show(cursor, get)),
        '</table>',
    ]

    cursor.close()
    connection.close()

    return '\n'.join(table)

app.run(host='0.0.0.0', threaded=True)
```

Introduction to Computer Science

Introduction to Computer Science
jerome.white@nyu.edu

jerome-white

Learning computer science concepts through practice.