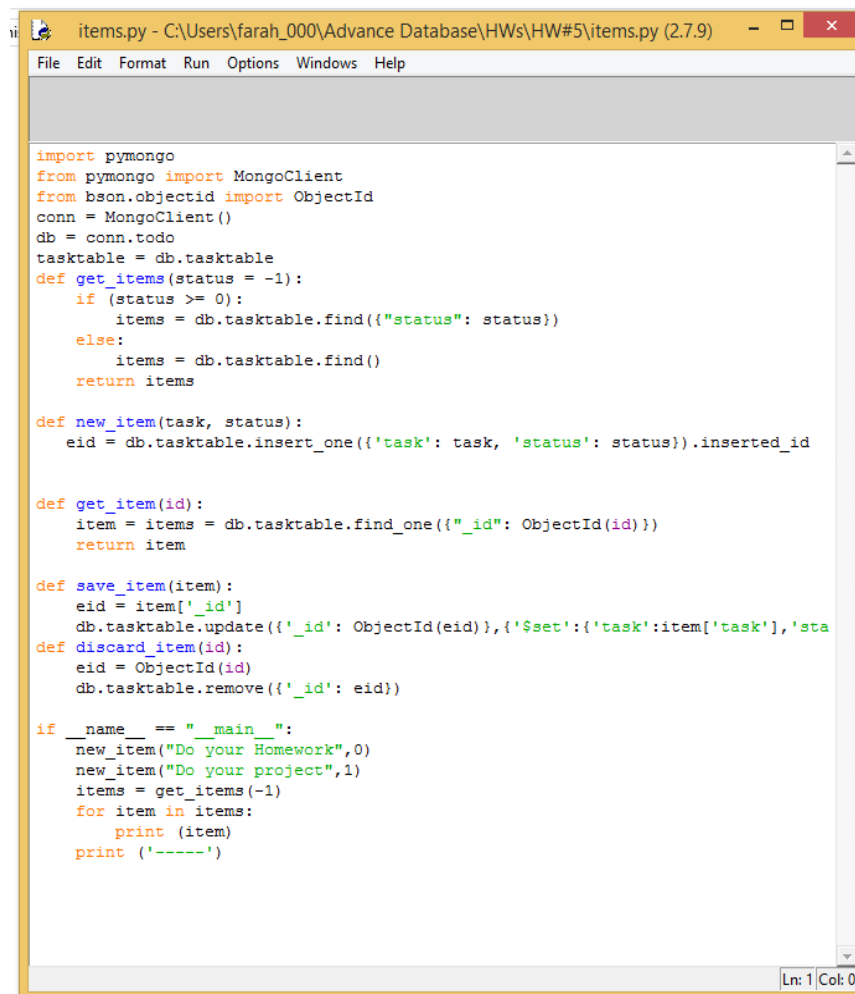


1. I read two tutorials about mongoDB and pymongo in the following two links
(http://www.tutorialspoint.com/mongodb/mongodb_create_database.htm)
(<http://api.mongodb.org/python/current/tutorial.html>)
2. I did the modification on your to do application (app.py) to make it work with mongoDB to insert, delete, update, retrieve items to/from the database. See the following figures for the code and the execution of the code. See the following link for the code

https://github.com/farahshleemon/Advanced-DataBase_HWs/tree/master/HW%235



```
items.py - C:\Users\farah_000\Advance Database\HWs\HW#5\items.py (2.7.9)
File Edit Format Run Options Windows Help

import pymongo
from pymongo import MongoClient
from bson.objectid import ObjectId
conn = MongoClient()
db = conn.todo
tasktable = db.tasktable
def get_items(status = -1):
    if (status >= 0):
        items = db.tasktable.find({"status": status})
    else:
        items = db.tasktable.find()
    return items

def new_item(task, status):
    eid = db.tasktable.insert_one({'task': task, 'status': status}).inserted_id

def get_item(id):
    item = items = db.tasktable.find_one({"_id": ObjectId(id)})
    return item

def save_item(item):
    eid = item['_id']
    db.tasktable.update({'_id': ObjectId(eid)}, {'$set': {'task': item['task'], 'sta

def discard_item(id):
    eid = ObjectId(id)
    db.tasktable.remove({'_id': eid})

if __name__ == "__main__":
    new_item("Do your Homework",0)
    new_item("Do your project",1)
    items = get_items(-1)
    for item in items:
        print (item)
    print ('-----')
```

```
*app.py - C:\Users\tarah_000\Advance Database\HWs\HW#5\app.py (2.7.9)*
File Edit Format Run Options Windows Help

from bottle import default_app, run, route
from bottle import get, put, post, request, template
from bottle import static_file
from items import *
@get('/')
@get('/list')
@get('/list/<status:int>')
def get_list(status=-1):
    items = get_items(status)
    result = []
    for item in items:
        result.append((item['id'], item['task'], item['status']))
    return template('list_view', rows=result)
@get('/new')
def get_new():
    return '''
    <p>Enter a new item...</p><br/>
    <form action="/new" method="post">
        To be done: <input name="task" type="text" />
        <input value="Save" type="submit" />
    </form>
    '''
@post('/new')
def post_new():
    task = request.forms.get('task', '').strip()
    new_item(task, 1)
    return get_list()
@get('/edit/<id>')
def get_edit(id):
    item = get_item(str(id))
    return template('edit_view', id=id, task=item['task'], status=item['status'])
@post('/edit/<id>')
def post_edit(id):
    task = request.forms.get('task', '').strip()
    status = request.GET.get('status', '').strip()
    if status == 'open':
        status = 1
    else:
        status = 0
    item = get_item(str(id))
    item['task'] = task
    item['status'] = status
    save_item(item)
    return get_list()
@get('/delete/<id>')
def delete_item(id):
    item = get_item(str(id))
    discard_item(str(id))
    return get_list()

if __name__ == '__main__':
    run(reloader=True, host='0.0.0.0')
else:
    application = default_app()

Ln: 31 Col: 81
11:02 PM
11/13/2015
```

```
*app.py - C:\Users\tarah_000\Advance Database\HWs\HW#5\app.py (2.7.9)*
File Edit Format Run Options Windows Help

def get_new():
    return '''
    <p>Enter a new item...</p><br/>
    <form action="/new" method="post">
        To be done: <input name="task" type="text" />
        <input value="Save" type="submit" />
    </form>
    '''
@post('/new')
def post_new():
    task = request.forms.get('task', '').strip()
    new_item(task, 1)
    return get_list()
@get('/edit/<id>')
def get_edit(id):
    item = get_item(str(id))
    return template('edit_view', id=id, task=item['task'], status=item['status'])
@post('/edit/<id>')
def post_edit(id):
    task = request.forms.get('task', '').strip()
    status = request.GET.get('status', '').strip()
    if status == 'open':
        status = 1
    else:
        status = 0
    item = get_item(str(id))
    item['task'] = task
    item['status'] = status
    save_item(item)
    return get_list()
@get('/delete/<id>')
def delete_item(id):
    item = get_item(str(id))
    discard_item(str(id))
    return get_list()
@get('/static/<filename>')
def server_static(filename):
    print(filename)
    return static_file(filename, root='static')

if __name__ == '__main__':
    run(reloader=True, host='0.0.0.0')
else:
    application = default_app()

Ln: 31 Col: 81
11:02 PM
11/13/2015
```

localhost:8080/new

Tutorial — PyMongo 3.1

MongoDB - Create Data

localhost:8080/new

Model List View

The open items are as follows:

#5646b39d0eed3329ecc7f106	Do HW's		
#5646b310eed3329ecc7f107	Do work on research		
#5646b3e50eed3329ecc7f108	Take some fun with family		
#5646b4050eed3329ecc7f10a	Teach Angel how to read		
#5646b4270eed3329ecc7f10a	Do more stuff		
#5646b4340eed3329ecc7f10b	do more fun thing		
#5646b47a0eed3329ecc7f10c	Read the documents for the paper		

Enter a new item...

To be done: Save new item...

localhost:8080/delete/564

Tutorial — PyMongo 3.1

MongoDB - Create Data

localhost:8080/delete/5646b39d0eed3329ecc7f106

Delete the task with ID = 5646b39d0eed3329ecc7f106

Task to delete:

Do HW's

Confirm deletion...

localhost:8080/edit/5646

Tutorial — PyMongo 3.1

MongoDB - Create Data

localhost:8080/edit/5646b4270eed3329ecc7f10a

Edit the task with ID = 5646b4270eed3329ecc7f10a

Do more stuff

open

save