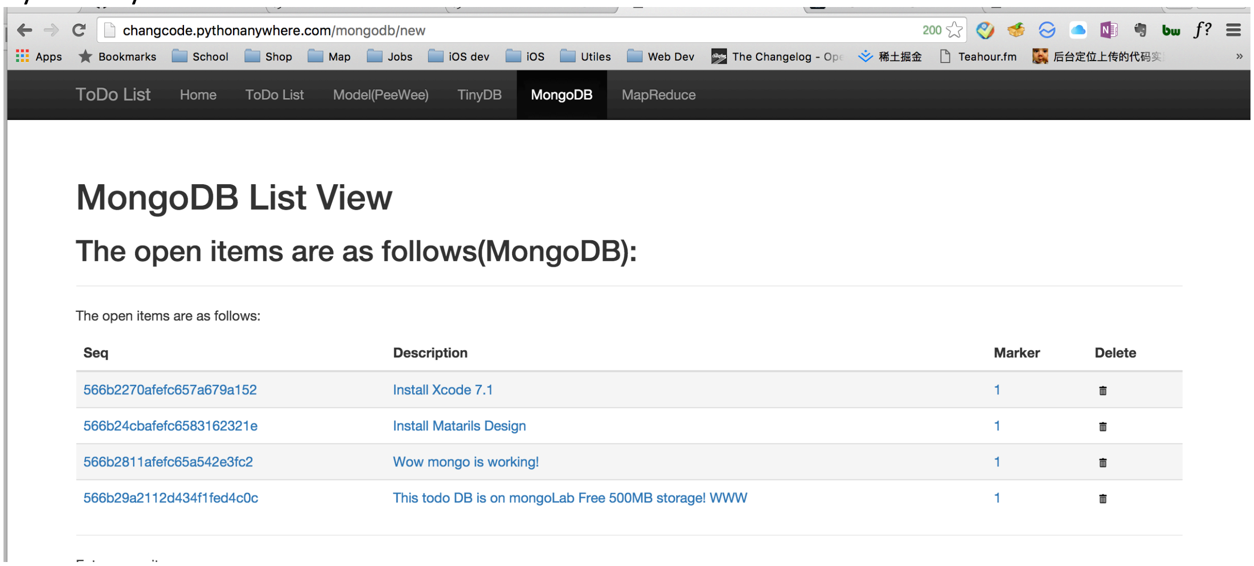
ADSD HW8 MongoDB ToDo

In this assignment I will MongoLab online mongo database as storage. It provide free 500MB space for mongo db.

You can visit from changcode.pythonanywhere.com to test CRUD operations.

Pythonanywhere:



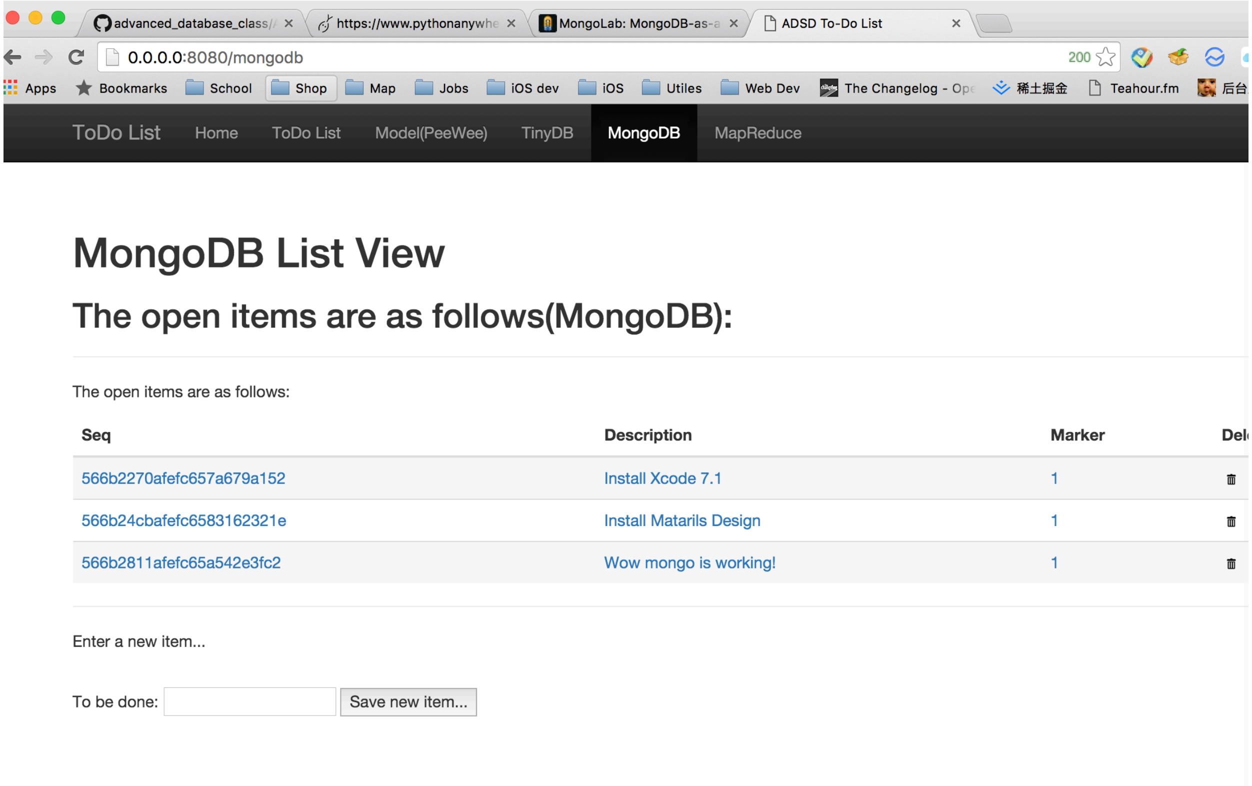
MongoDB List View

The open items are as follows(MongoDB):

The open items are as follows:

Seq	Description	Marker	Delete
566b2270afefc657a679a152	Install Xcode 7.1	1	🗑
566b24cbafefc6583162321e	Install Matarils Design	1	🗑
566b2811afefc65a542e3fc2	Wow mongo is working!	1	🗑
566b29a2112d434f1fed4c0c	This todo DB is on mongoLab Free 500MB storage! WWW	1	🗑

Local:



MongoDB List View

The open items are as follows(MongoDB):

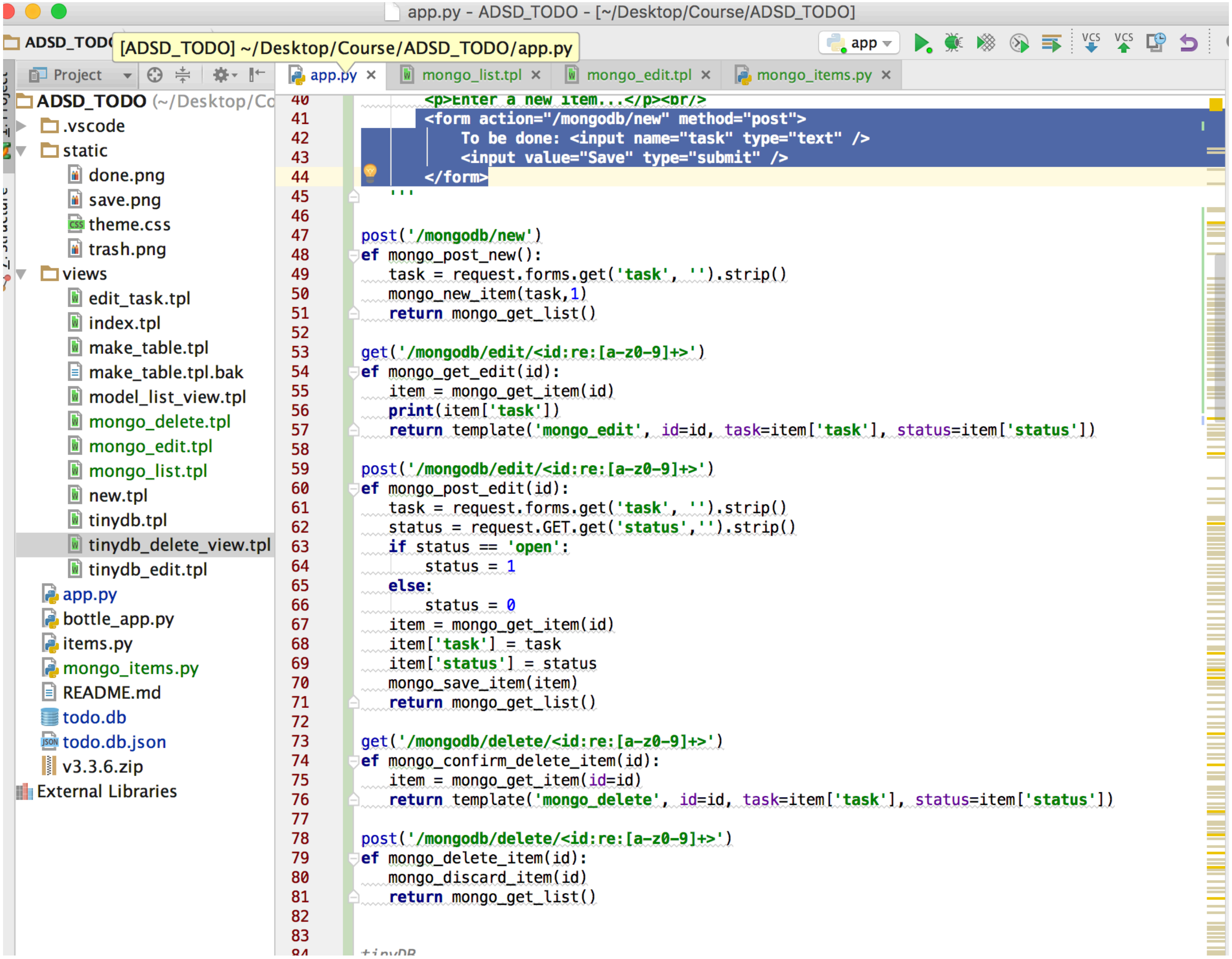
The open items are as follows:

Seq	Description	Marker	Del
566b2270afefc657a679a152	Install Xcode 7.1	1	🗑
566b24cbafefc6583162321e	Install Matarils Design	1	🗑
566b2811afefc65a542e3fc2	Wow mongo is working!	1	🗑

Enter a new item...

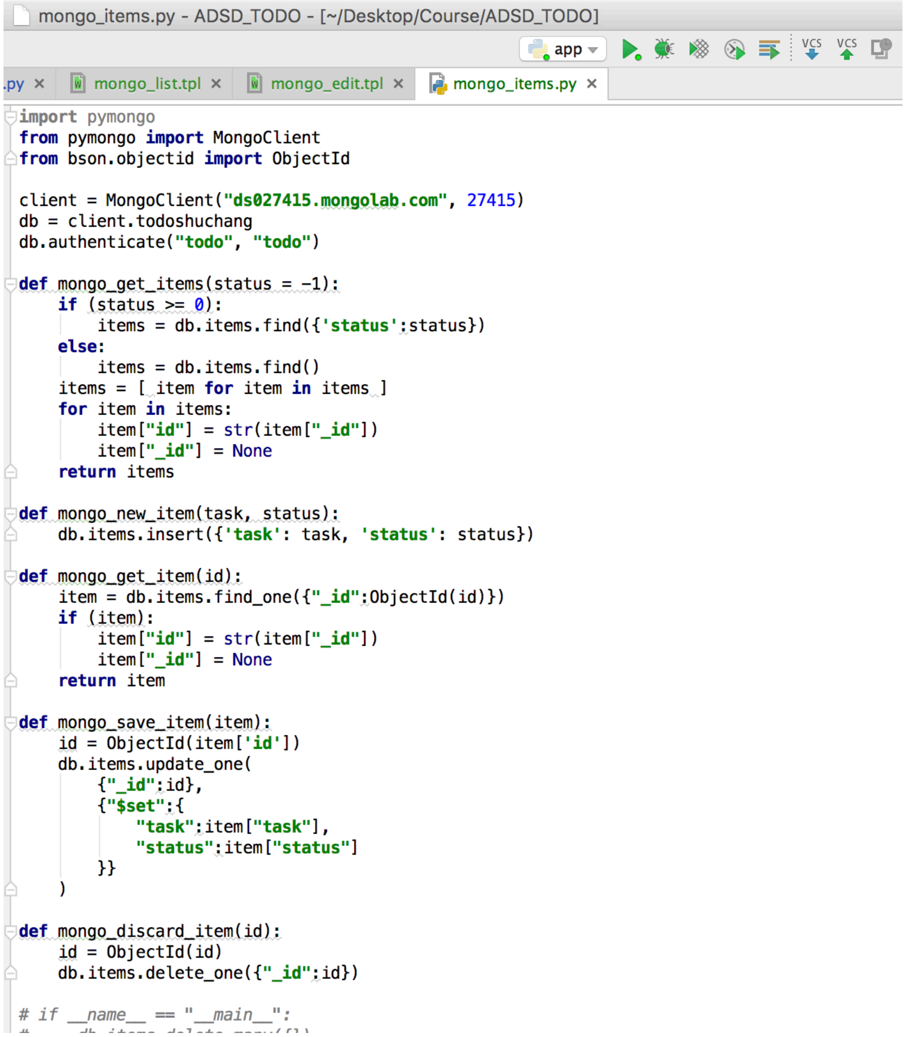
To be done: [ ] Save new item...

Code:



```
       <p>Enter a new item...</p><br/>
       <form action="/mongodb/new" method="post">
           To be done: <input name="task" type="text" />
           <input value="Save" type="submit" />
       </form>
   '''

post('/mongodb/new')
ef mongo_post_new():
   task = request.forms.get('task', '').strip()
   mongo_new_item(task,1)
   return mongo_get_list()

get('/mongodb/edit/<id:re:[a-z0-9]+>')
ef mongo_get_edit(id):
   item = mongo_get_item(id)
   print(item['task'])
   return template('mongo_edit', id=id, task=item['task'], status=item['status'])

post('/mongodb/edit/<id:re:[a-z0-9]+>')
ef mongo_post_edit(id):
   task = request.forms.get('task', '').strip()
   status = request.GET.get('status','').strip()
   if status == 'open':
       status = 1
   else:
       status = 0
   item = mongo_get_item(id)
   item['task'] = task
   item['status'] = status
   mongo_save_item(item)
   return mongo_get_list()

get('/mongodb/delete/<id:re:[a-z0-9]+>')
ef mongo_confirm_delete_item(id):
   item = mongo_get_item(id=id)
   return template('mongo_delete', id=id, task=item['task'], status=item['status'])

post('/mongodb/delete/<id:re:[a-z0-9]+>')
ef mongo_delete_item(id):
   mongo_discard_item(id)
   return mongo_get_list()


tinyDB
```

MongoDBItems:



```python
import pymongo
from pymongo import MongoClient
from bson.objectid import ObjectId

client = MongoClient("ds027415.mongolab.com", 27415)
db = client.todoshuchang
db.authenticate("todo", "todo")

def mongo_get_items(status = -1):
    if (status >= 0):
        items = db.items.find({'status':status})
    else:
        items = db.items.find()
    items = [ item for item in items ]
    for item in items:
        item["id"] = str(item["_id"])
        item["_id"] = None
    return items

def mongo_new_item(task, status):
    db.items.insert({'task': task, 'status': status})

def mongo_get_item(id):
    item = db.items.find_one({"_id":ObjectId(id)})
    if (item):
        item["id"] = str(item["_id"])
        item["_id"] = None
    return item

def mongo_save_item(item):
    id = ObjectId(item['id'])
    db.items.update_one(
        {"_id":id},
        {"$set":{
            "task":item["task"],
            "status":item["status"]
        }}
    )

def mongo_discard_item(id):
    id = ObjectId(id)
    db.items.delete_one({"_id":id})

# if __name__ == "__main__":
#     db.items.delete_many({})
```