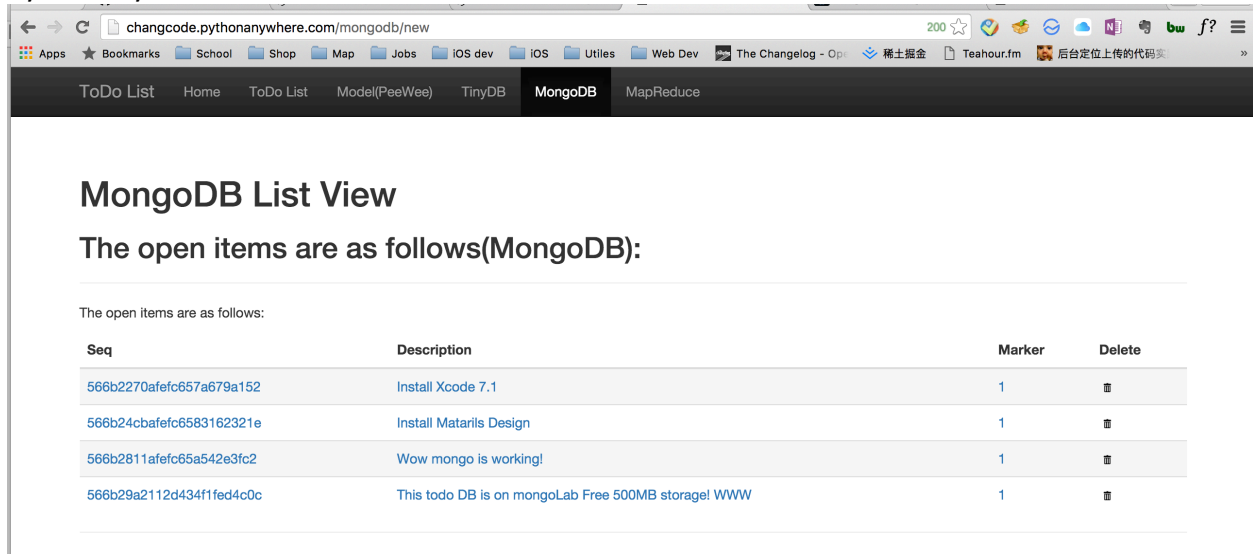


ADSD HW8 MongoDB ToDo

In this assignment I will use MongoDB online mongo database as storage. It provides free 500MB space for mongo db.

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

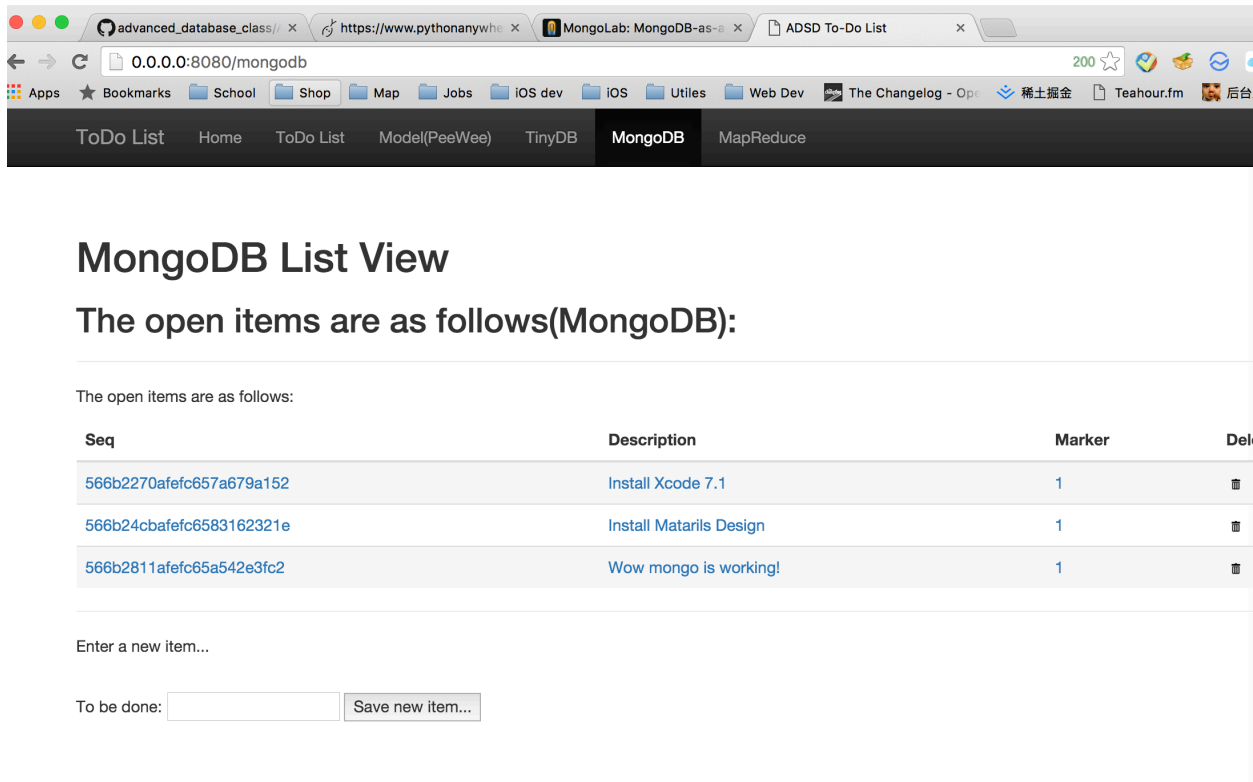
Pythonanywhere:



The screenshot shows a web browser window with the URL `changcode.pythonanywhere.com/mongodb/new`. The browser's address bar and tabs are visible. The page has a dark navigation bar with links: `ToDo List`, `Home`, `ToDo List`, `Model(PeeWee)`, `TinyDB`, `MongoDB` (active), and `MapReduce`. The main content area is titled **MongoDB List View** and contains the text "The open items are as follows(MongoDB):". Below this, a table lists four items:

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:



The screenshot shows a web browser window with the URL `0.0.0.0:8080/mongodb`. The browser's address bar and tabs are visible. The page has a dark navigation bar with links: `ToDo List`, `Home`, `ToDo List`, `Model(PeeWee)`, `TinyDB`, `MongoDB` (active), and `MapReduce`. The main content area is titled **MongoDB List View** and contains the text "The open items are as follows(MongoDB):". Below this, a table lists three items:

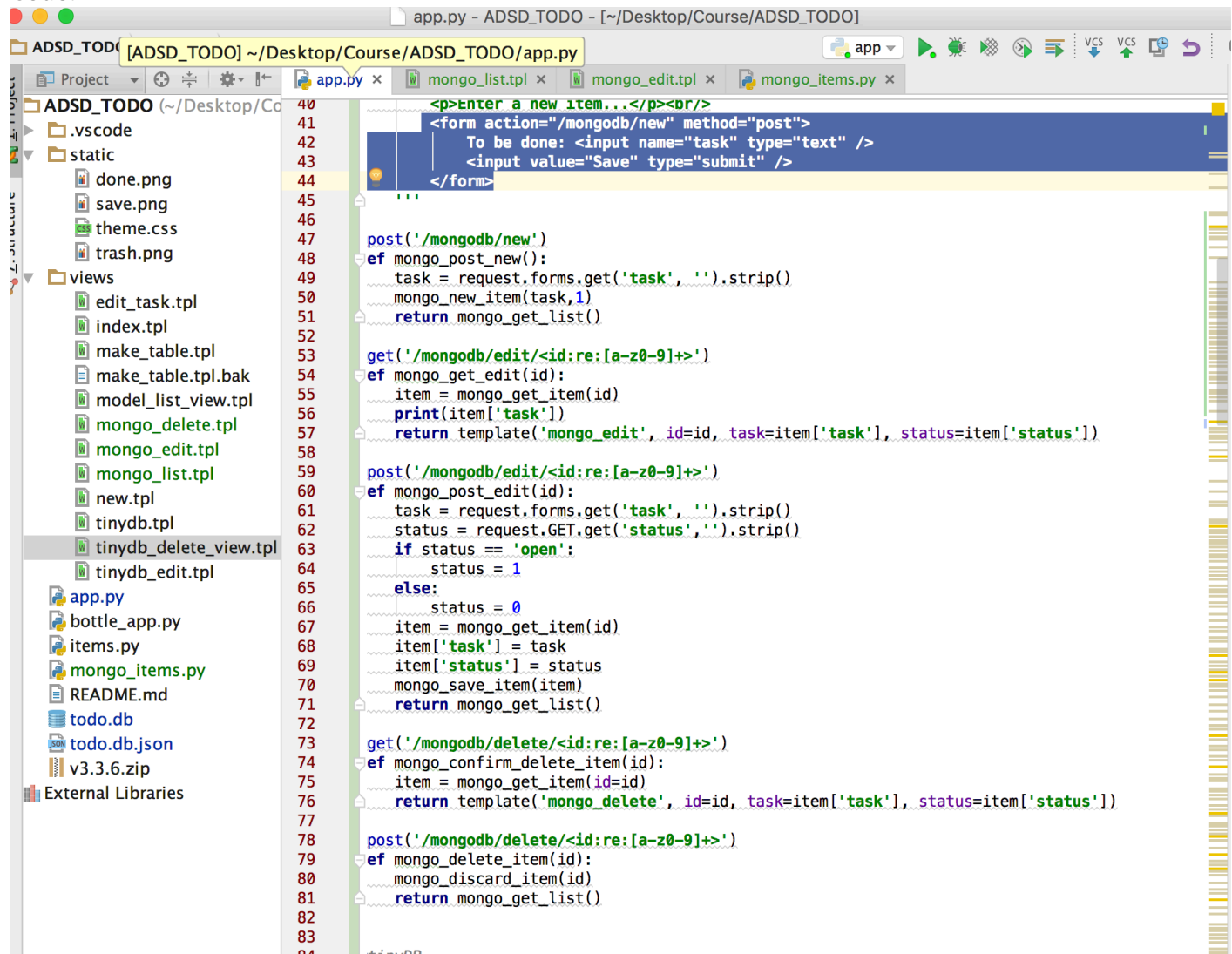
Seq	Description	Marker	Delete
566b2270afefc657a679a152	Install Xcode 7.1	1	
566b24cbafefc6583162321e	Install Matarils Design	1	
566b2811afefc65a542e3fc2	Wow mongo is working!	1	

Below the table, there is a form to add a new item:

Enter a new item...

To be done:

Code:



```
app.py - ADSD_TODO - [~/Desktop/Course/ADSD_TODO]
ADSD_TODO [ADSD_TODO] ~/Desktop/Course/ADSD_TODO/app.py
Project
ADSD_TODO (~/Desktop/Course/ADSD_TODO)
  .vscode
  static
    done.png
    save.png
    theme.css
    trash.png
  views
    edit_task.tpl
    index.tpl
    make_table.tpl
    make_table.tpl.bak
    model_list_view.tpl
    mongo_delete.tpl
    mongo_edit.tpl
    mongo_list.tpl
    new.tpl
    tinydb.tpl
    tinydb_delete_view.tpl
    tinydb_edit.tpl
  app.py
  bottle_app.py
  items.py
  mongo_items.py
  README.md
  todo.db
  todo.db.json
  v3.3.6.zip
  External Libraries

40 <p>Enter a new item...</p><br/>
41 <form action="/mongodb/new" method="post">
42   To be done: <input name="task" type="text" />
43   <input value="Save" type="submit" />
44 </form>
45 ...
46
47 post('/mongodb/new')
48 def mongo_post_new():
49     task = request.forms.get('task', '').strip()
50     mongo_new_item(task, 1)
51     return mongo_get_list()
52
53 get('/mongodb/edit/<id:re:[a-z0-9]+>')
54 def mongo_get_edit(id):
55     item = mongo_get_item(id)
56     print(item['task'])
57     return template('mongo_edit', id=id, task=item['task'], status=item['status'])
58
59 post('/mongodb/edit/<id:re:[a-z0-9]+>')
60 def mongo_post_edit(id):
61     task = request.forms.get('task', '').strip()
62     status = request.GET.get('status', '').strip()
63     if status == 'open':
64         status = 1
65     else:
66         status = 0
67     item = mongo_get_item(id)
68     item['task'] = task
69     item['status'] = status
70     mongo_save_item(item)
71     return mongo_get_list()
72
73 get('/mongodb/delete/<id:re:[a-z0-9]+>')
74 def mongo_confirm_delete_item(id):
75     item = mongo_get_item(id=id)
76     return template('mongo_delete', id=id, task=item['task'], status=item['status'])
77
78 post('/mongodb/delete/<id:re:[a-z0-9]+>')
79 def mongo_delete_item(id):
80     mongo_discard_item(id)
81     return mongo_get_list()
82
83
84
```

MongoDBItems:

```
mongo_items.py - ADSD_TODO - [~/Desktop/Course/ADSD_TODO]
app
.py x mongo_list.tpl x mongo_edit.tpl x mongo_items.py x
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__":
#     # TODO: Add your code here
```