**Assignment**

1. **REST Web API:**

Using Nodejs REST WEB API to communicate on webservers connected to MongoDB server.

Important code to check for the REST operations available in routes/users.js

**GET Operation:** Data of student information is getting saved in **users/userlist**.

router.get('/userlist', function(req, res) {

var db = req.db;

var collection = db.get('userlist');

collection.find({},{},function(e,docs){

res.json(docs);

});

});

**POST Operation:**

router.post('/adduser', function(req, res) {

var db = req.db;

var collection = db.get('userlist');

collection.insert(req.body, function(err, result){

res.send(

(err === null) ? { msg: '' } : { msg: err }

);

});

});

**Delete Operation:**

router.delete('/deleteuser/:id', function(req, res) {

var db = req.db;

var collection = db.get('userlist');

var userToDelete = req.params.id;

collection.remove({ '\_id' : userToDelete }, function(err) {

res.send((err === null) ? {msg: '' } : { msg:'error: ' + err });

});

});

module.exports = router;

1. **Interact With API (Python scripts to perform CRUD Operations)**

**GET: (Extracting the data from database)**

import urllib, json

url = <http://10.103.45.50:3000/users/userlist>

response = urllib.urlopen(url)

data = json.loads(response.read())

index = -1

for i in range(len(a)):

if a[i]['email']==email:

index = i

break

if index == -1:

print "Not found"

print "Student Details: ",data

**POST: (Adding/uploading data)**

import requests

import json

payload = {}

#create an empty dictionary

url = "http://10.103.45.50:3000/users/adduser"

#Insert the data in dictionary

payload['username'] = str(raw\_input("Enter username : "))

payload['fullname'] = str(raw\_input("Enter fullname : "))

payload['email'] = str(raw\_input("Enter email : "))

payload['age'] = str(raw\_input("Enter username : "))

payload['location'] = str(raw\_input("Enter username : "))

payload['gender'] = str(raw\_input("Enter username : "))

r = requests.post(url, data=json.dumps(payload))

print(r.text)

**Delete: (Removing data)**

import urllib, json

url = "http://10.103.45.50:3000/users/userlist"

response = urllib.urlopen(url)

data = json.loads(response.read())

# removing the user information by taking it's email-id as input

def db\_search(a):

email = raw\_input("Enter the email id of user:\n")

index = -1

for i in range(len(a)):

if a[i]['email']==email:

index = i

break

if index == -1:

print "Not found"

a.remove(a[index])

print "Old database: ",data

db\_search(data)

print "New database: ",data

1. **Scale Up application with NGINX:**

**Load Balancing: /etc/nginx/site-available**

1. Install NGINX: sudo apt­get install ­y nginx

upstream web\_backend {

    server 10.0.0.15;

    server 10.0.0.16;

}

server {

    listen 80;

    location / {

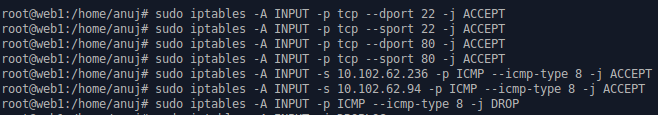
        proxy\_set\_header X­Forwarded­For $proxy\_add\_x\_forwarded\_for;

        proxy\_pass http://web\_backend;

    }

}

1. **IP Tables:**

****

1. **Automation with Ansible**

**Ansible playbook to install Nginx on web servers:**

**/etc/ansible/ansible-playbook.yml**

---

- host: all

sudo: True

tasks:

- name: Install Nginx

apt : pkg=nginx state=present

handlers:

- name: start nginx

service: name=nginx state=started

