College Database

# Project Details

**Write few lines of details about the business problem you are trying to implement or solve.**

College database contains the data of students, instructors, courses, classes, departments and divisions.

The user can easily access the data of any student and instructor etc. from anywhere once it is deployed to cloud .Any change has been made to database easily.

# GitHub URL

<https://github.com/Simranjeet-123>

# Team Members:

|  |  |
| --- | --- |
| Student ID | Student Name |
| 744302 | Simranjeet Kaur |

# NoSQL Database Information

Database Name: CollegeDatabase

|  |  |  |
| --- | --- | --- |
| Collection Name | Collection Details and #of Documents available | Team Member(s) worked on it |
| Student | Student collection contains information of students such as name, phone number etc.  # of documents:8 | Simranjeet Kaur |
| Instructor | Instructor collection contains information of instructor such as name, classes etc.  # of documents:8 | Simranjeet Kaur |
| Course | Course collection contains the description of Courses such as name etc.  # of documents:8 | Simranjeet Kaur |
| Class | Course collection contains the information about classes such as class type, courses taught etc.  # of documents:8 | Simranjeet Kaur |
| Department | Department collection contains the information about classes such as class type, courses taught etc.  # of documents:8 | Simranjeet Kaur |
| Division | Division collection contains the information about Division such as name, department id etc.  # of documents:3 | Simranjeet Kaur |
|  |  |  |

Database backup file: Attach your database backup file here so that it can be restored and used.

# NodeJS Service Information

|  |  |  |
| --- | --- | --- |
| Service Path | Details | Team Member(s) worked on it |
| localhost:8081/api/student | Get service to retrieve all the documents in student collection of mongo database. | Simranjeet Kaur |
| Localhost:8081/api/instructor | Get service to retrieve all the documents in instructor collection of mongo database. | Simranjeet Kaur |
| localhost:8081 /api/student1 | Post service to insert data into student collection. | Simranjeet Kaur |
| localhost:8081 /api/instructor1 | Post service to insert data into instructor collection. | Simranjeet Kaur |
| localhost:8081 /api/student2 | Put service to update student name in student collection | Simranjeet Kaur |
| localhost:8081 /api/instructor2 | Put service to update instructor name in instructor collection | Simranjeet Kaur |
| localhost:8081/api/deleteStudent | Delete service to delete one document from student collection. | Simranjeet Kaur |
| localhost:8081/api/student/:student\_id | Get service to retrieve a particular document using student\_id as a parameter in student collection of mongo database. | Simranjeet Kaur |
| localhost:8081/api/instructor/:instructor\_name | Get service to retrieve a particular document using instructor\_name as a parameter in instructor collection. | Simranjeet Kaur |

Service Codes: Attach the NodeJS code package here.

const express = require('express');

const app = express();

app.use(express.json());

//welcome page

app.get('/',(req,res)=>{

res.send('welcome to mongonodejs page');

});

// show all the collections in mongo db

app.get('/api/show\_collections',(req,res)=>{

const mongo = require('mongodb');

const MongoClient = mongo.MongoClient;

const url = 'mongodb://localhost:27017';

MongoClient.connect(url, { useNewUrlParser: true }, (err, client) => {

if (err) throw err;

const db = client.db("collegeDatabase");

db.listCollections().toArray().then((docs) => {

console.log('Available collections:');

docs.forEach((doc, idx, array) => { console.log(doc.name) });

}).catch((err) => {

console.log(err);

}).finally(() => {

client.close();

});

});

});

// get service to retrieve student collection data

app.get('/api/student',(req,res)=>{

var MongoClient = require('mongodb').MongoClient;

var url ="mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {

if (err) throw err;

var dbo = db.db("collegeDatabase");

dbo.collection("student").find({}).toArray( function(err, result) {

if (err) throw err;

console.log(result);

res.send(result);

db.close();

});

});

});

// get service to retrieve instructor collection data

app.get('/api/instructor',(req,res)=>{

var MongoClient = require('mongodb').MongoClient;

var url ="mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {

if (err) throw err;

var dbo = db.db("collegeDatabase");

dbo.collection("instructor").find({}).toArray( function(err, result) {

if (err) throw err;

res.send(result);

db.close();

});

});

});

// post service to insert data into student collection data

app.post('/api/student1',(req,res)=>{

var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {

if (err) throw err;

var dbo = db.db("collegeDatabase");

var myobj = { student\_id:"209",student\_name: "Mahir",student\_phone:"9876541234",student\_address:"Toronto",student\_email:"mahir@gmail",courses\_taken:[103,105]};

dbo.collection("student").insertOne(myobj, function(err, res) {

if (err) throw err;

console.log("1 document inserted into student collection");

});

});

});

// post service to insert data into instructor collection data

app.post('/api/instructor1',(req,res)=>{

var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {

if (err) throw err;

var dbo = db.db("collegeDatabase");

var myobj = { instructor\_id: "408",instructor\_name: "Mehar Sharma",course\_id:[104,102]};

dbo.collection("instructor").insertOne(myobj, function(err, res) {

if (err) throw err;

console.log("1 document inserted into instructor collection");

});

});

});

// put service to update data into student collection data

app.put('/api/student2',(req,res)=>{

var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://127.0.0.1:27017/";

MongoClient.connect(url, function(err, db) {

if (err) throw err;

var dbo = db.db("collegeDatabase");

var myquery = { student\_name: "Mahir" };

var newvalues = { $set: {student\_id: "209", student\_name:"Mahir Malik" } };

dbo.collection("student").updateOne(myquery, newvalues, function(err, res) {

if (err) throw err;

console.log("1 document updated");

db.close();

});

});

});

// put service to update data in instructor collection data

app.put('/api/instructor2',(req,res)=>{

var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://127.0.0.1:27017/";

MongoClient.connect(url, function(err, db) {

if (err) throw err;

var dbo = db.db("collegeDatabase");

var myquery = { instructor\_name: "Mehar Sharma" };

var newvalues = { $set: {student\_id: "408", instructor\_name:"Mehar" } };

dbo.collection("instructor").updateOne(myquery, newvalues, function(err, res) {

if (err) throw err;

console.log("1 document updated");

db.close();

});

});

});

app.delete('/api/deleteStudent',(req,res)=>{

var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {

if (err) throw err;

var dbo = db.db("collegeDatabase");

var myquery = { student\_id:"209" };

dbo.collection("student").deleteOne(myquery, function(err, obj) {

if (err) throw err;

console.log("1 document deleted from student collection");

res.send("1 document deleted from student collection")

db.close();

});

});

});

app.delete('/api/deleteInstructor',(req,res)=>{

var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {

if (err) throw err;

var dbo = db.db("collegeDatabase");

var myquery = { instructor\_id: "408" };

dbo.collection("instructor").deleteOne(myquery, function(err, obj) {

if (err) throw err;

res.send("1 document deleted from instructor collection");

db.close();

});

});

});

// Get using parameter

app.get('/api/student/:student\_id', (req, res) => {

var MongoClient = require('mongodb').MongoClient;

var url ="mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {

if (err) throw err;

var dbo = db.db("collegeDatabase");

dbo.collection("student").find({}).toArray( function(err, result) {

if (err) throw err;

console.log(result);

const student1=result;

const stu = student1.find( c=> c.student\_id === (req.params.student\_id));

if(!stu) res.status(404).send('<h2 style="font-family: Malgun Gothic; color: darkred;"> Not Found!! </h2>');

res.send(stu);

});

});

});

// Get using parameter

app.get('/api/instructor/:instructor\_name', (req, res) => {

var MongoClient = require('mongodb').MongoClient;

var url ="mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {

if (err) throw err;

var dbo = db.db("collegeDatabase");

dbo.collection("instructor").find({}).toArray( function(err, result) {

if (err) throw err;

console.log(result);

const instructor1=result;

const inst = instructor1.find( c=> c.instructor\_name === (req.params.instructor\_name));

if(!inst) res.status(404).send('<h2 style="font-family: Malgun Gothic; color: darkred;"> Not Found!! </h2>');

res.send(inst);

});

});

});

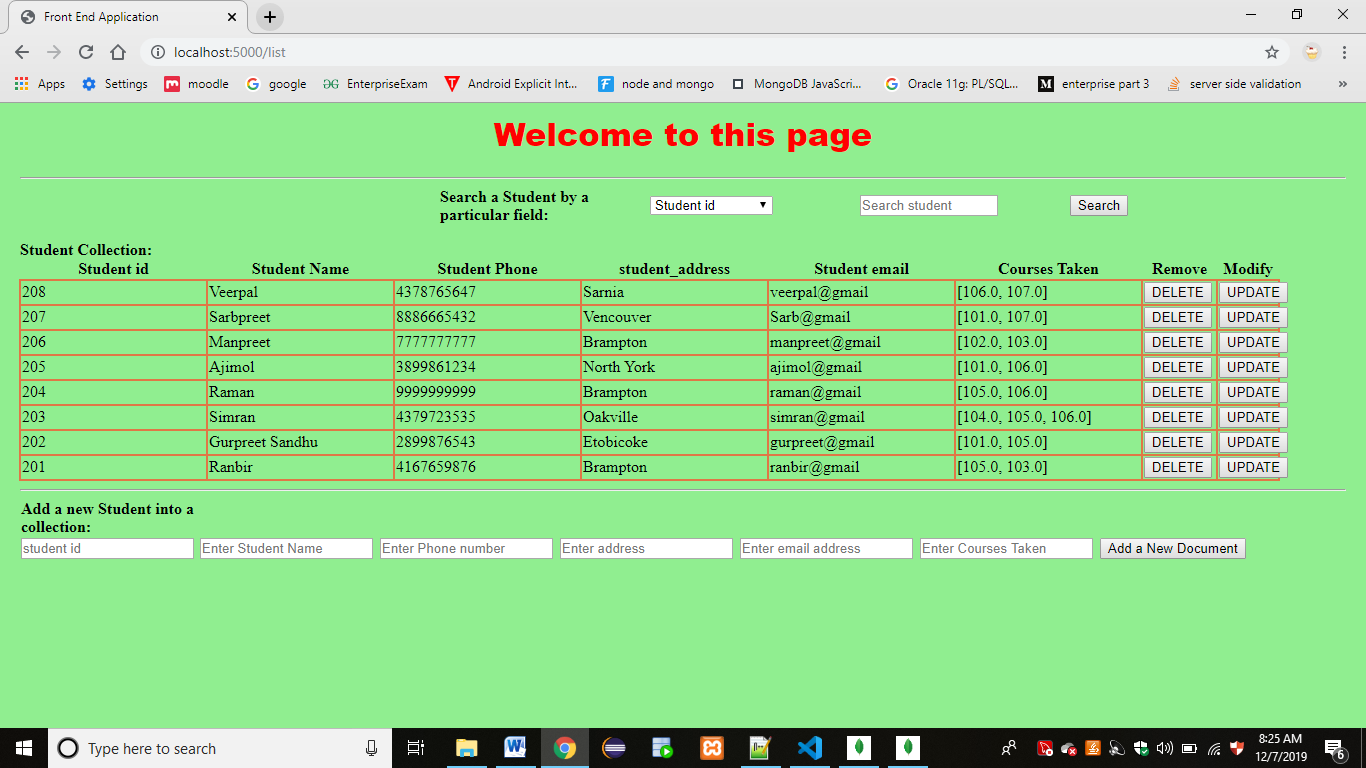
const port=process.env.PORT || 8085;

app.listen(port,() => console.log(`listening on port ${port}..`));

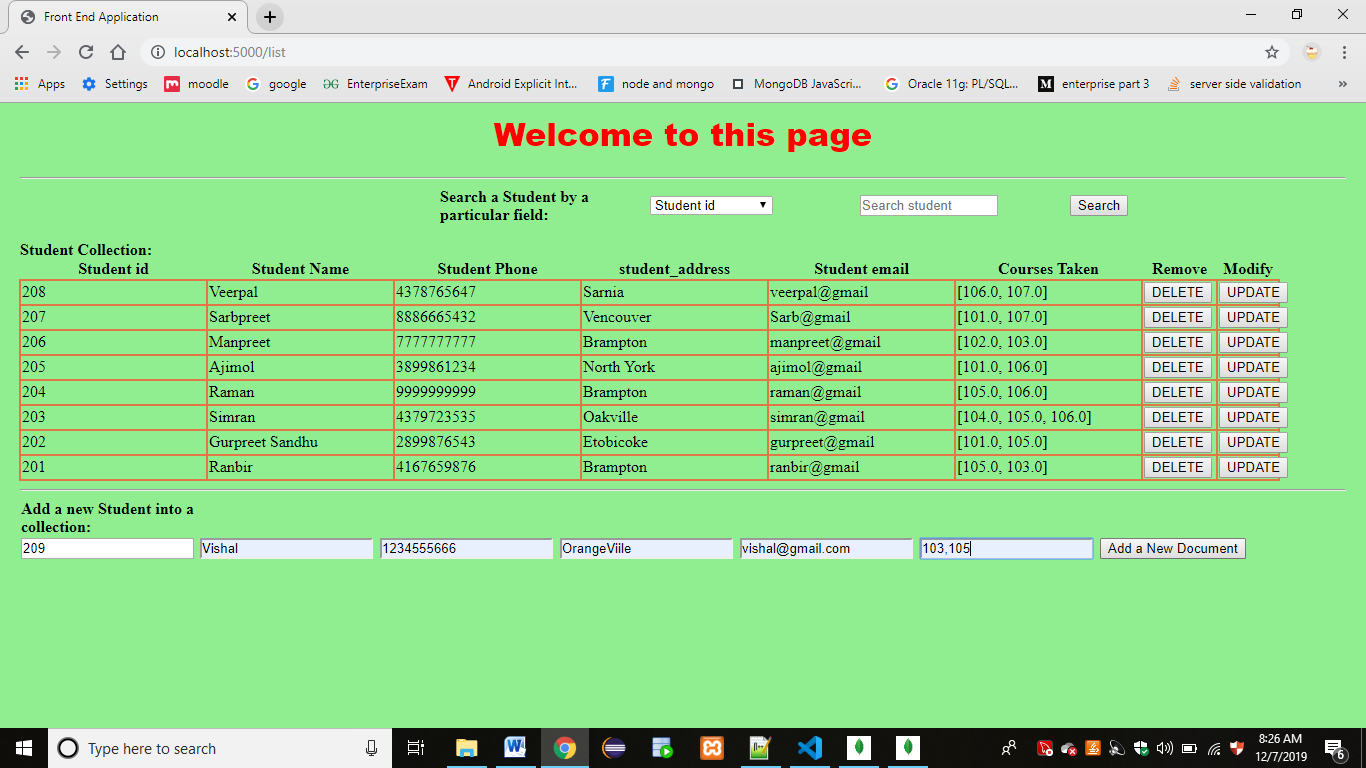
# Application or Website details

Provide screen shot with details.

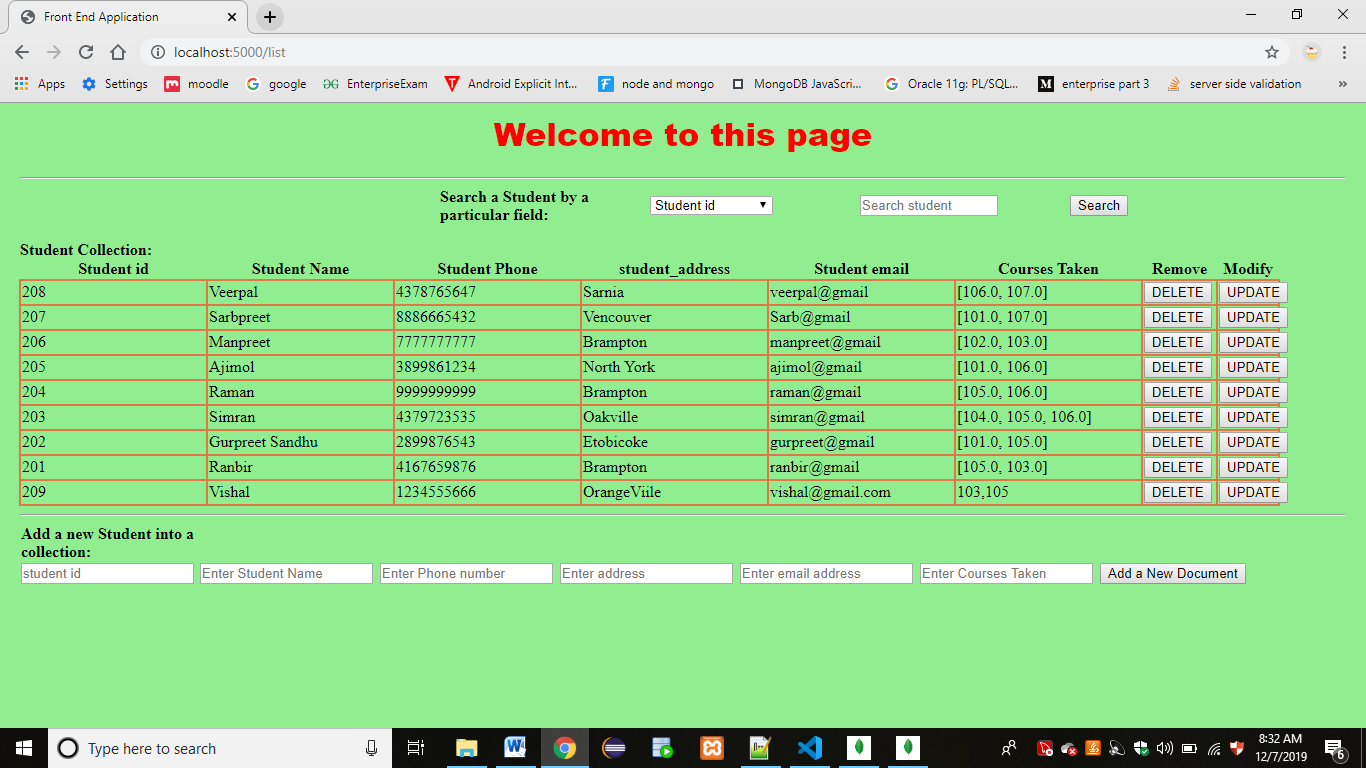
Index.html: Main Page browser



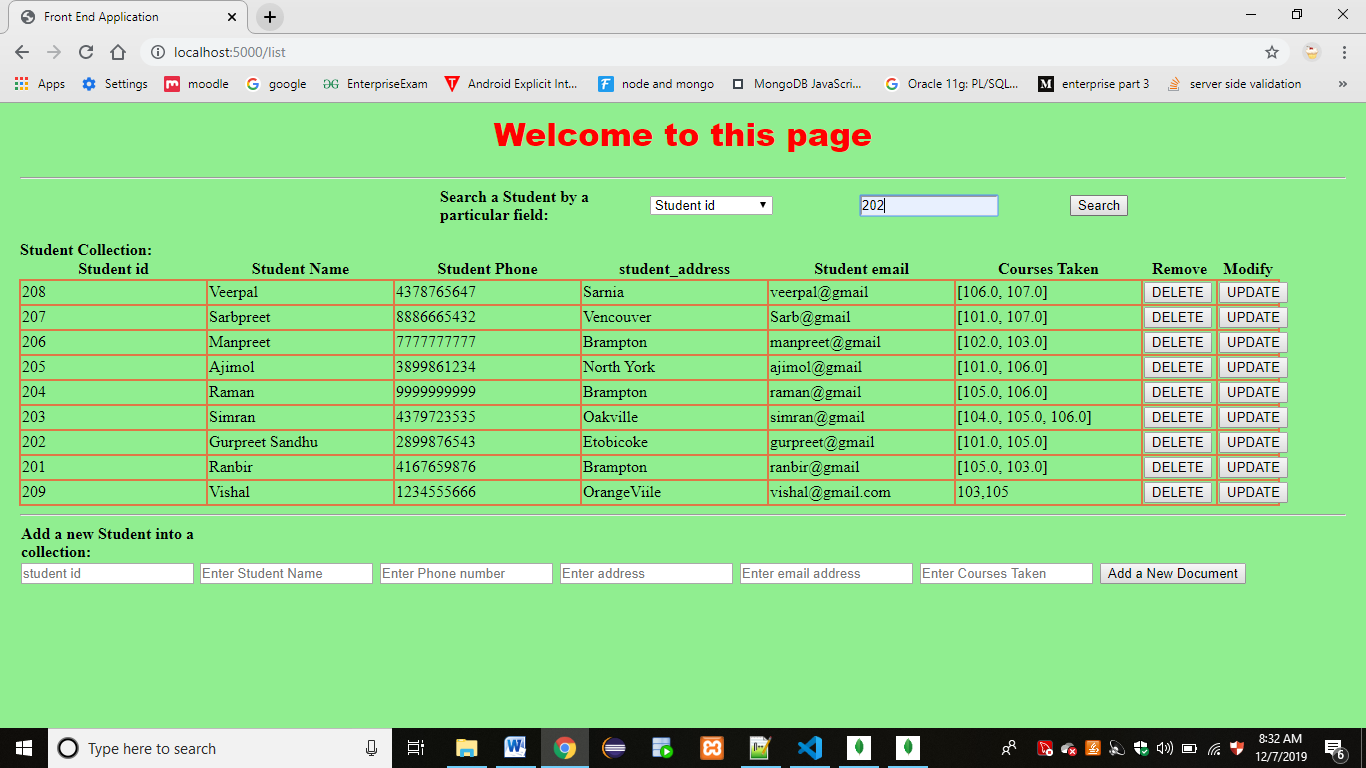
Output to add a new document:



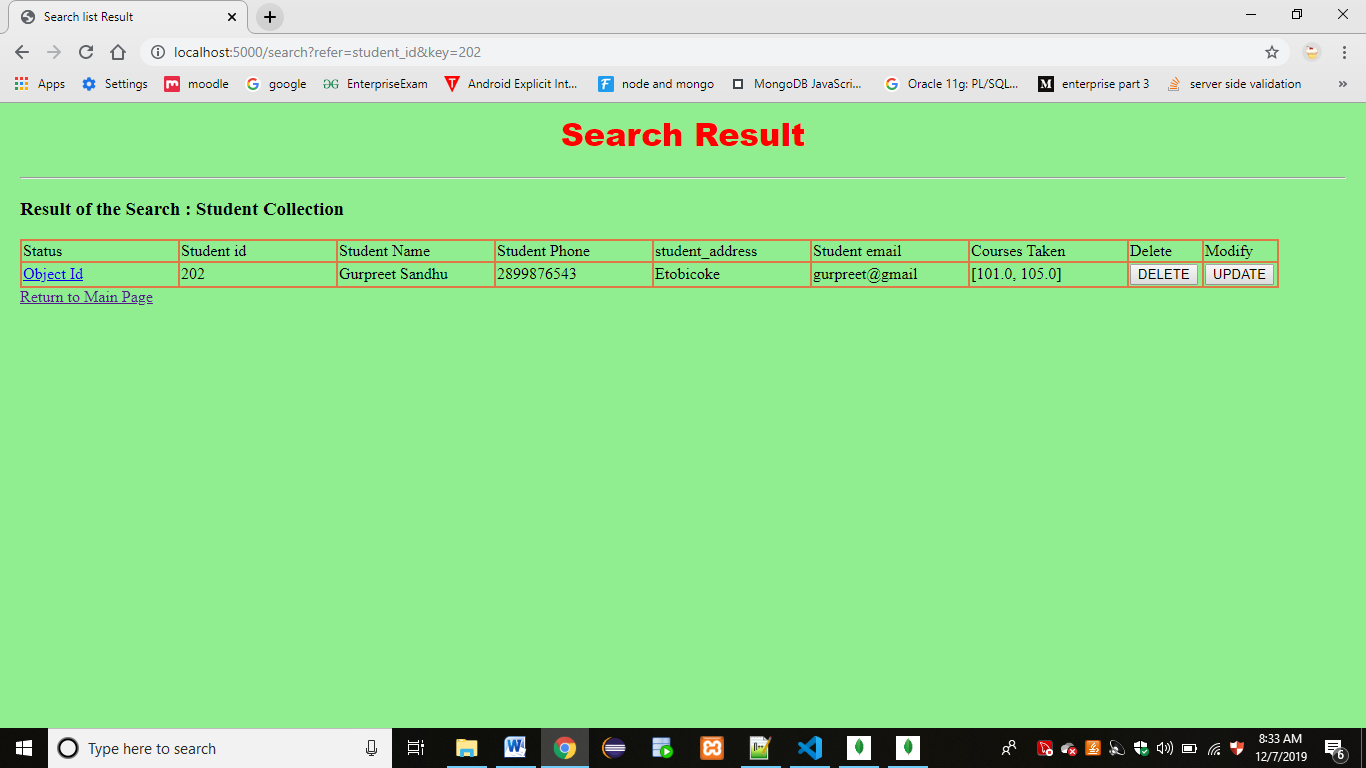
AFTER CLICKING ADD A NEW DOCUMENT BUTTON



SEARCH A DOCUMENT BY REFERENCE:

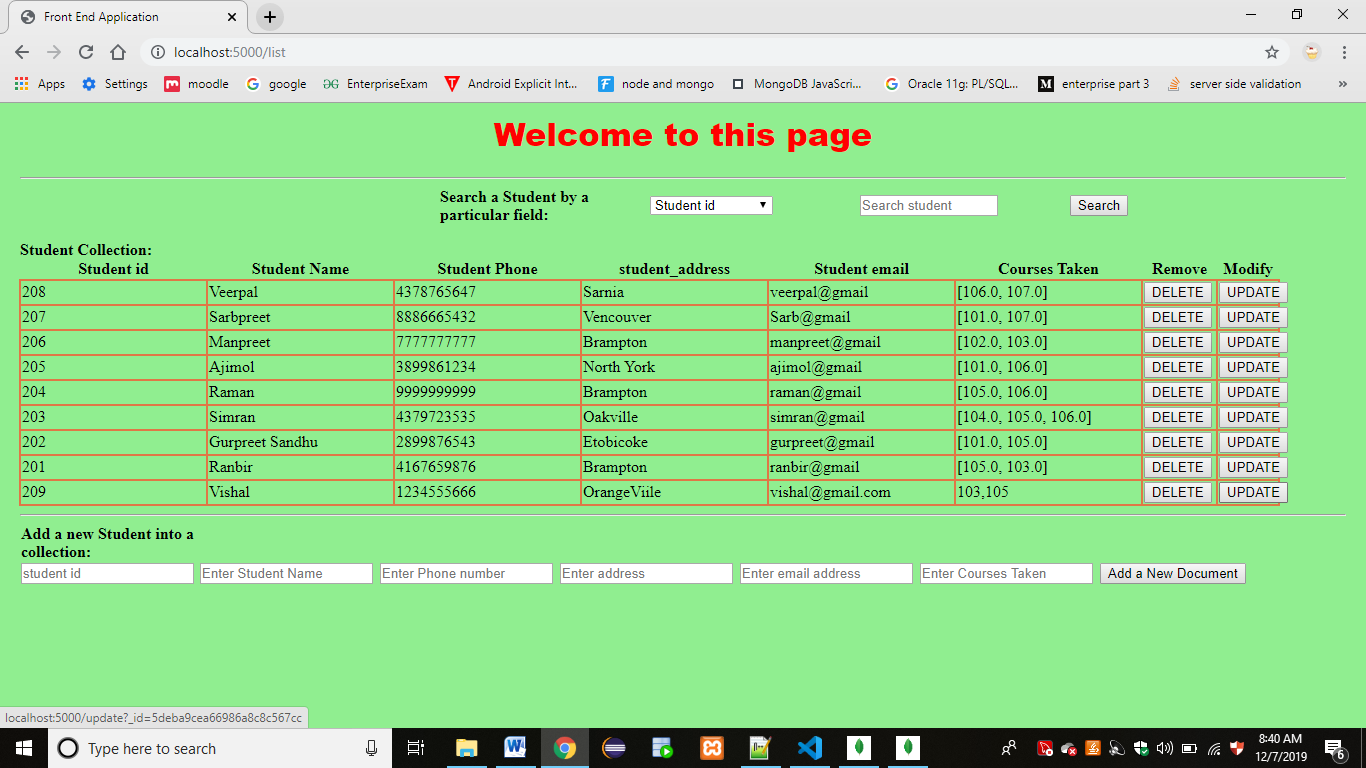


SEARCHLIST.HTML

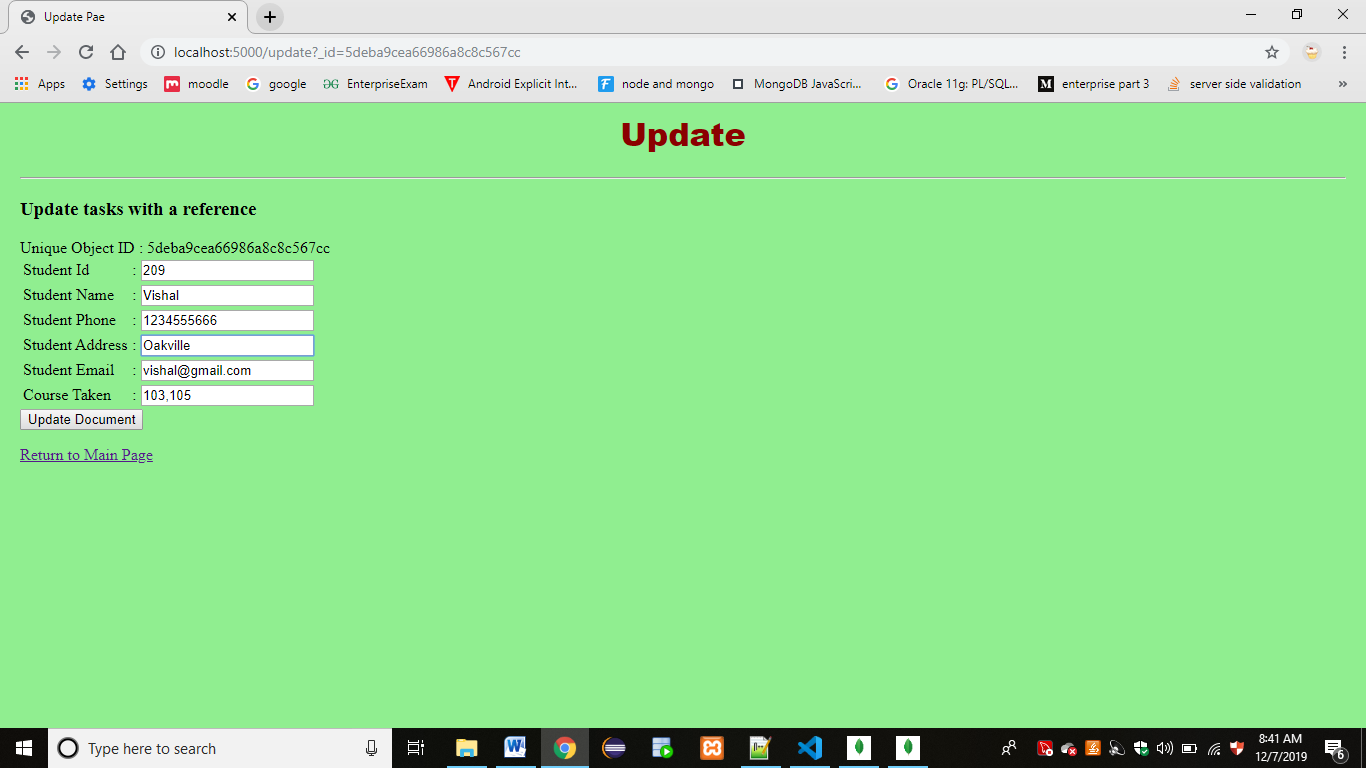


Update a document:

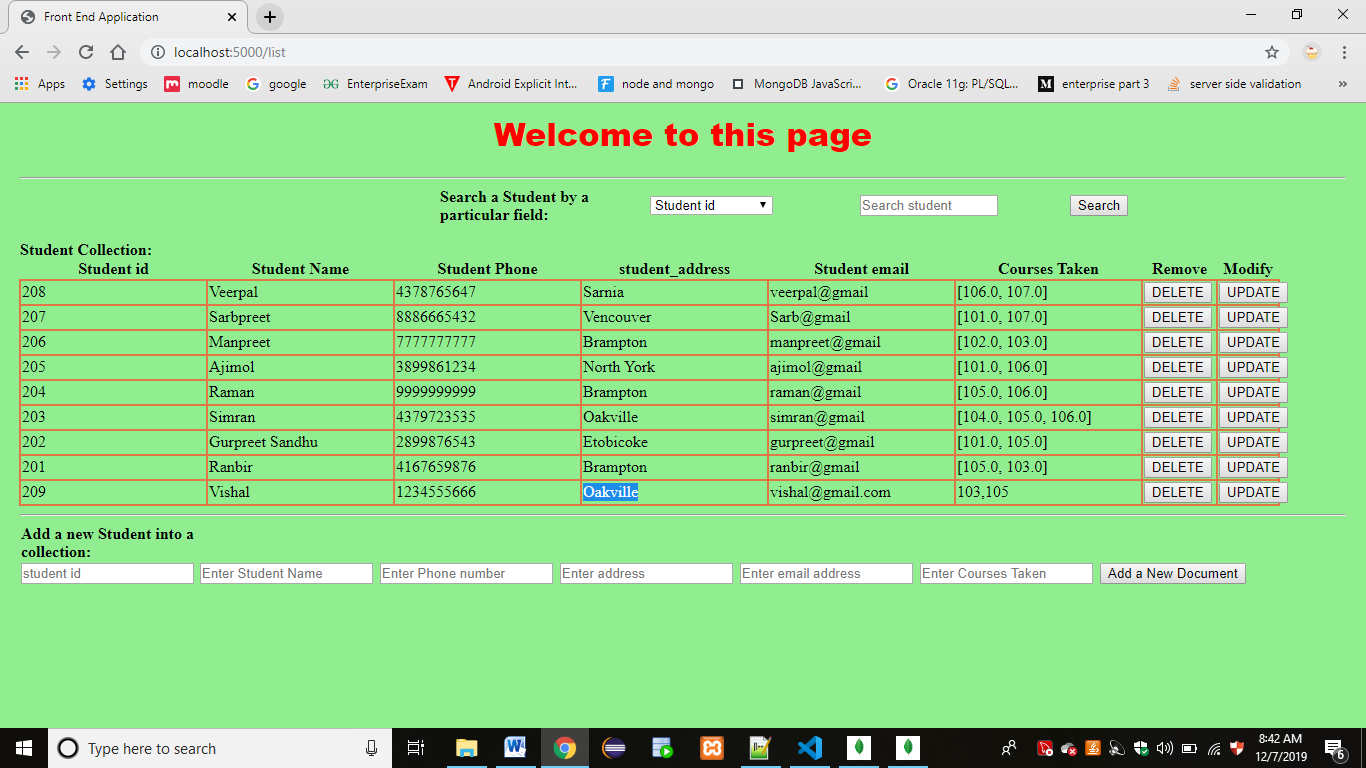
Click on update button for student id 209 and it will take you to update.html page



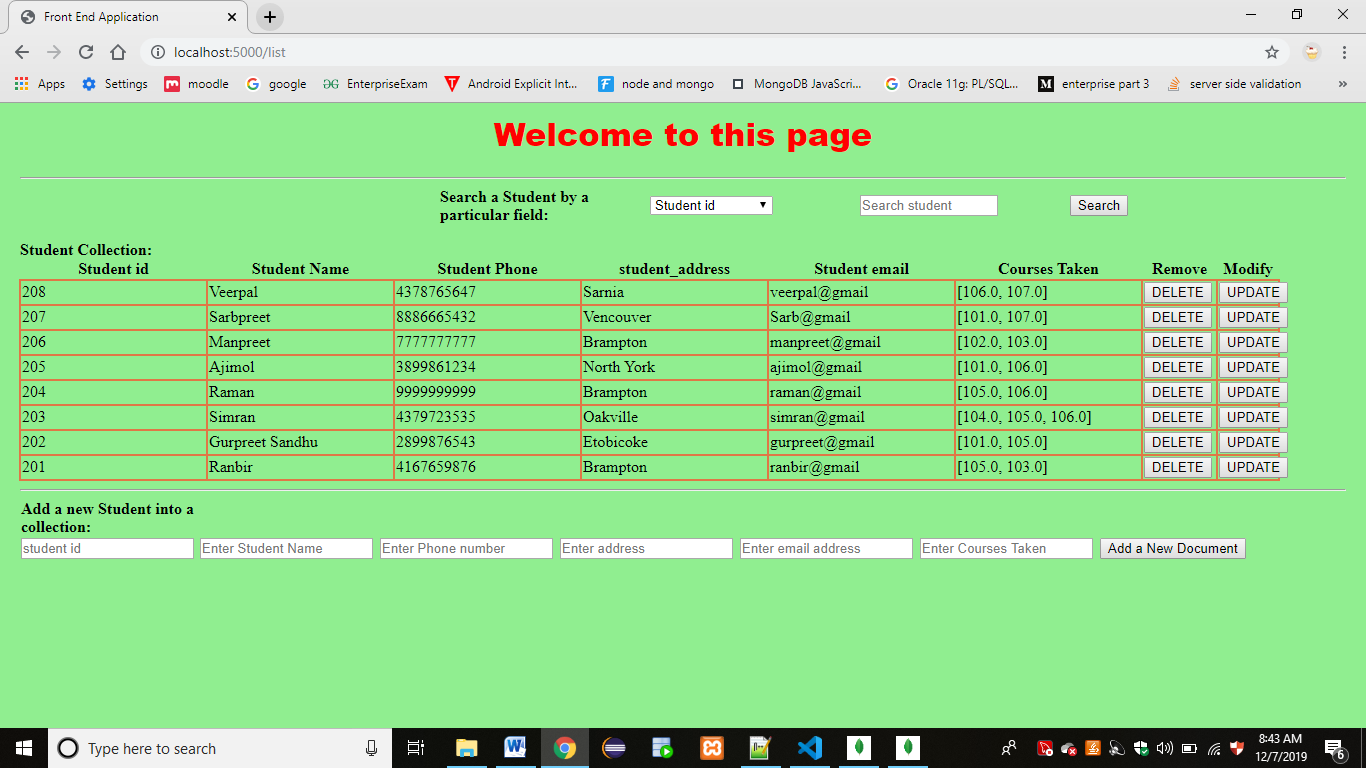
I updated address to Oakville from Orangeville:



Click on update document button:



Delete a document 209 by clicking on delete button:



Attach the codes.

Python File:app.py

from flask import Flask, render\_template,request,redirect,url\_for # For flask implementation

from bson import ObjectId # For ObjectId to work

from pymongo import MongoClient

import os

app = Flask(\_\_name\_\_)

title = " Student Collection with Flask and MongoDB"

heading = "Get,put,post and delete in student collection"

client = MongoClient("mongodb://127.0.0.1:27017") #host uri

db = client.collegeDatabase    #Select the database

students = db.student #Select the collection name

def redirect\_url():

    return request.args.get('next') or \

           request.referrer or \

           url\_for('index')

@app.route("/list")

def lists ():

    #Display the all Tasks

    student\_1 = students.find()

    a1="active"

    return render\_template('index.html',a1=a1,students=student\_1,t=title,h=heading)

@app.route("/insert", methods=['POST'])

def action ():

    #Adding a Task

    stu\_id=request.values.get("student\_id")

    stu\_name=request.values.get("student\_name")

    stu\_phone=request.values.get("student\_phone")

    stu\_address=request.values.get("student\_address")

    stu\_email=request.values.get("student\_email")

    cour\_taken=request.values.get("courses\_taken")

    students.insert({ "student\_id":stu\_id, "student\_name":stu\_name, "student\_phone":stu\_phone,"student\_address":stu\_address, "student\_email":stu\_email,"courses\_taken":cour\_taken})

    return redirect("/list")

@app.route("/remove")

def remove ():

    #Deleting a student with various references

    key=request.values.get("\_id")

    students.remove({"\_id":ObjectId(key)})

    return redirect("/list")

@app.route("/update")

def update ():

    id=request.values.get("\_id")

    student\_1=students.find({"\_id":ObjectId(id)})

    return render\_template('update.html',tasks=student\_1,h=heading,t=title)

@app.route("/action3", methods=['POST'])

def action3 ():

    stu\_id=request.values.get("id")

    stu\_name=request.values.get("name")

    stu\_phone=request.values.get("phone")

    stu\_address=request.values.get("address")

    stu\_email=request.values.get("email")

    cour\_taken=request.values.get("courses")

    id=request.values.get("\_id")

    students.update({"\_id":ObjectId(id)}, {'$set':{ "student\_id":stu\_id, "student\_name":stu\_name, "student\_phone":stu\_phone, "student\_address":stu\_address, "student\_email":stu\_email,"courses\_taken":cour\_taken }})

    return redirect("/list")

@app.route("/search", methods=['GET'])

def search():

    #Searching a Task with various references

    key=request.values.get("key")

    refer=request.values.get("refer")

    if(key=="\_id"):

        student\_l = students.find({refer:ObjectId(key)})

    else:

        student\_l = students.find({refer:key})

    return render\_template('searchlist.html',students=student\_l,t=title,h=heading)

if \_\_name\_\_ == "\_\_main\_\_":

    app.run()

Main Page: index.html

<html>

    <head>

        <title>Front End Application</title>

        <style>

             h1{

                    font-family:"Arial Black", Gadget, sans-serif;

                    color:red;

                    text-align: center;

                }

            body{

                    color:black;

                    background-color:lightgreen;

                    margin-left:20px;

                    margin-right:20px;

                }

            table{

                    width:95%;

                    border-collapse:collapse;

                    table-layout:fixed;

                }

            td

                {

                    border: rgba(224, 119, 70, 1) 2px solid;

                    word-wrap:break-word;

                }

            table#close{

                    table-layout:default;

                    border-collapse:seperate;

                    border-spacing: 5px 5px;

                    border:none;

                    vertical-align:top;

                }

            table.none

                {

                    border:none;

                    vertical-align:top;

                }

            table.none td

                {

                    border: none;

                }

            tr.row td{

                    text-decoration:italic;

                }

            th.status{

                    width:5%;

                }

            th.name{

                    width:20%;

                }

            th.des{

                    width:40%;

                    padding:5px 5px 5px 5px;

                }

            th.date{

                    width:8%;

                }

            th.pr{

                    width:7%;

                }

            th.func1{

                    width:6%;

                }

            th.func2{

                    width:5%;

                }

            input[type=submit] {

                    width: 20em;  height: 2em;

                }

            ul {

                    list-style-type: none;

                    margin: 0;

                    padding: 0;

                    overflow: hidden;

                    background-color: #333;

                }

            li {

                    float: left;

                }

            li a {

                    display: block;

                    color: white;

                    text-align: center;

                    padding: 14px 16px;

                    text-decoration: none;

                }

            a.active {

                    background-color: #4CAF50;

                }

                /\* Change the link color to #111 (black) on hover \*/

            li a:hover {

                    background-color:royalblue;

                }

        </style>

    </head>

<body>

    <h1>Welcome to this page</h1>

    <hr>

    {% if students[0] %}

    <div span="right">

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

        <table class="none" id="close">

        <tr>

        <td></td><td></td>

        <td><b>Search a Student by a particular field:</b></td>

        <td><select name="refer" >

            <option value="student\_id">Student id</option>

            <option value="student\_name">Student Name</option>

            <option value="student\_phone">Student Phone</option>

            <option value="student\_address">Student Address </option>

            <option value="student\_email">Student Email</option>

            <option value="courses\_taken">Courses Taken</option>

        </select></td>

        <td><input type="text" name="key" placeholder="Search student" size="15" /></td>

        <td><button type="submit">Search</button></td>

        </tr>

        </table>

    </form>

    </div>

    <b>Student Collection:</b>

    <table>

        <tr id="row">

            <th class="student\_id">Student id</th>

            <th class="student\_name">Student Name</th>

            <th class="student\_phone">Student Phone</th>

            <th class="student\_address">student\_address</th>

            <th class="student\_email">Student email</th>

            <th class="courses\_taken">Courses Taken</th>

        <th class="func1">Remove</th>

        <th class="func2">Modify</th>

        </tr>

    {% for stu in students  %}

        <tr class="datas">

            <td class="student\_id">{{ stu["student\_id"] }}</td>

            <td class="student\_name">{{ stu["student\_name"] }}</td>

            <td class="student\_phone">{{ stu["student\_phone"] }}</td>

            <td class="student\_address">{{ stu["student\_address"] }}</td>

            <td class="student\_email">{{ stu["student\_email"] }}</td>

            <td class="courses\_taken">{{ stu["courses\_taken"] }}</td>

            <td class="func1"><a href="./remove?\_id={{ stu['\_id'] }}"><button type="submit">DELETE</button></a></td>

            <td class="func1"><a href="./update?\_id={{ stu['\_id'] }}"><button type="submit">UPDATE</button></a></td>

        </tr>

    {% endfor %}

    </table>

    {% else %}

        <h4>No document in Student Collection !!</h4>

    {% endif %}

    <hr/>

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

    <table class="none">

        <tr>

            <td><b><label>Add a new Student into a collection: </label></b></td>

        </tr>

        <tr>

        <td><input type="text" name="student\_id" placeholder="student id" /></td>

        <td><input  type="text" name="student\_name"  placeholder="Enter Student Name"></td>

        <td><input type="text" name="student\_phone" placeholder="Enter Phone number" /></td>

        <td><input type="text" name="student\_address" placeholder="Enter address" /></td>

        <td><input type="text" name="student\_email" placeholder="Enter email address" /></td>

        <td><input type="text" name="courses\_taken" placeholder="Enter Courses Taken" /></td>

        <td><button type="submit">Add a New Document</button></td>

        </tr>

    </table>

    </form>

</body>

</html>

Update.html

<html>

    <head>

        <title>Update Pae</title>

        <style>

                h1{

                    font-family:"Arial Black", Gadget, sans-serif;

                    text-align: center;

                    color: darkred;

                }

                body{

                    color:black;

                    background-color:lightgreen;

                    margin-left:20px;

                    margin-right:20px;

                }

            </style>

    </head>

<body>

    <h1>Update</h1>

    <hr>

    <h3>Update tasks with a reference</h3>

    <form action="/action3" method="POST">

    {% for stu in tasks %}

        Unique Object ID : {{ stu['\_id'] }}<br/>

    <input type="text" name="\_id" value="{{ stu['\_id'] }}" hidden>

    <table>

    <tr>

        <td>Student Id</td>

        <td> : </td>

        <td><input type="text" name="id" value="{{ stu['student\_id'] }}" placeholder="{{  stu['student\_id'] }}"></td>

    </tr>

    <tr>

        <td>Student Name</td>

        <td> : </td>

        <td><input type="text" name="name" value="{{ stu['student\_name'] }}" placeholder="{{  stu['student\_name'] }}"></td>

    </tr>

    <tr>

        <td>Student Phone</td>

        <td> : </td>

        <td><input type="text" name="phone" value="{{ stu['student\_phone'] }}" placeholder="{{  stu['student\_phone'] }}"></td>

    </tr>

    <tr>

        <td>Student Address</td>

        <td> : </td>

        <td><input type="text" name="address" value="{{ stu['student\_address'] }}" placeholder="{{  stu['student\_address'] }}"></td>

    </tr>

    <tr>

        <td>Student Email</td>

        <td> : </td>

        <td><input type="text" name="email" value="{{ stu['student\_email'] }}" placeholder="{{  stu['student\_email'] }}"></td>

    </tr>

    <tr>

        <td>Course Taken</td>

        <td> : </td>

        <td><input type="text" name="courses" value="{{ stu['courses\_taken'] }}" placeholder="{{  stu['courses\_taken'] }}"></td>

    </tr>

    </table>

    {% endfor %}

        <button type="submit"> Update Document </button>

        <br/>

    </form>

    <a href="/list">Return to Main Page</a>

</body>

</html>

Searchlist.html

<html>

    <head>

        <title>Search list Result</title>

        <style>

            h1{

                    font-family:"Arial Black", Gadget, sans-serif;

                    /\*font-family:"Times New Romans", Gadget, sans-serif;\*/

                    color:red;

                    text-align: center;

                }

            body{

                    color:black;

                    background-color:lightgreen;

                    margin-left:20px;

                    margin-right:20px;

                }

            table{

                    width:95%;

                    border-collapse:collapse;

                    table-layout:fixed;

                }

            td

                {

                    border: rgba(224, 119, 70, 1) 2px solid;

                    word-wrap:break-word;

                }

            table#close{

                    table-layout:default;

                    border-collapse:seperate;

                    border-spacing: 5px 5px;

                    border:none;

                    vertical-align:top;

                }

            table.none

                {

                    border:none;

                    vertical-align:top;

                }

            table.none td

                {

                    border: none;

                }

            tr.row td{

                    text-decoration:italic;

                }

            th.status{

                    width:5%;

                }

            th.name{

                    width:20%;

                }

            th.des{

                    width:40%;

                    padding:5px 5px 5px 5px;

                }

            th.date{

                    width:8%;

                }

            th.pr{

                    width:7%;

                }

            td.func{

                    width:6%;

                }

            td.func1{

                    width:5%;

                }

            input[type=submit] {

                    width: 20em;  height: 2em;

                }

        </style>

    </head>

<body>

    <h1>Search Result</h1>

    <hr>

    {% if students[0] %}

    <h3>Result of the Search : Student Collection</h3>

    <table>

        <tr id="row">

            <td class="status">Status</td>

            <td class="student\_id">Student id</td>

            <td class="student\_name">Student Name</td>

            <td class="student\_phone">Student Phone</td>

            <td class="student\_address">student\_address</td>

            <td class="student\_email">Student email</td>

            <td class="courses\_taken">Courses Taken</td>

        <td class="func">Delete</td>

        <td class="func">Modify</td>

        </tr>

        {% for stu in students %}

        <tr class="datas">

            <td><a href="./done?\_id={{stu['\_id'] }}">Object Id</a></td>

            <td class="student\_id">{{ stu["student\_id"] }}</td>

            <td class="student\_name">{{ stu["student\_name"] }}</td>

            <td class="student\_phone">{{ stu["student\_phone"] }}</td>

            <td class="student\_address">{{ stu["student\_address"] }}</td>

            <td class="student\_email">{{ stu["student\_email"] }}</td>

            <td class="courses\_taken">{{ stu["courses\_taken"] }}</td>

            <td class="func1"><a href="./remove?\_id={{ stu['\_id'] }}"><button type="submit">DELETE</button></a></td>

            <td class="func1"><a href="./update?\_id={{ stu['\_id'] }}"><button type="submit">UPDATE</button></a></td>

        </tr>

        {% endfor %}

    {% else %}

        <h4>No Result Found !!</h4>

    {% endif %}

    </table>

    <a href="/list">Return to Main Page</a>

</body>

</html>

# How this application will be deployed into Cloud?

**DEPLOY A WEB APPLICATION TO CLOUD**

* **We can use AWS (Amazon Web services) cloud environment to deploy all the files (python application) to cloud so that it can be accessible from anywhere.**
* **Firstly I Log in to aws.amazon.com.**
* **There are multiple services that we can choose from. To run python script I choose EC2.**
* **To launch an EC2 service, I will launch an instance also called virtual server.**
* **Select a system that has python installed in it. Amazon Linux Has python installed, so I create an instance of it.**
* **It will ask to create a new key pair name. I will create A key pair, then a key pair name (for example, simran.pem ) file will be downloaded.**
* **Once we create an instance, then instance state will be changed to running.**
* **I click on connect.**
* **I will select a Region and bits etc. needed to upload my application.**
* **Once I upload my files and fill all the required fields, My application will be deployed to AWS CLOUD Environment.**
* **I can use Paas or Iaas .But best one is Paas because I am dealing with only my applications and database and other infrastructure settings will be handled by the cloud server operator, in this**

**AWS is responsible for network and other infrastructure.**

# Score Distribution

# Score Distribution

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module | Total Marks | Student 1 | Student 2 | Student 3 | Student 4 |
| NoSQL Database | 8 |  |  |  |  |
| NodeJS Service | 8 |  |  |  |  |
| Application | 8 |  |  |  |  |
| Cloud Deployment Details | 8 |  |  |  |  |
| GitHub | 4 |  |  |  |  |
| Participation | 4 |  |  |  |  |
| Total | 40 |  |  |  |  |