COS 301 PHASE 3 – GROUP 11

MERLOT FACIAL RECOGNITION SUBSYSTEM

MEET THE TEAM

EMMA COETZER, HERBERT MAGAYA, KOKETSO MOLAWA, JUSTIN ROSS

EMMA COETZER



Responsible for creating the MongoDB database to store the client's photographs used for the facial recognition. I created methods to interact with the database. Also responsible for the landing page and the PowerPoint presentation.

HERBERT MAGAYA

Responsible for communicating with the rest of Merlot teams, most specifically with the authentication team. Also implemented the UPDATE function responsible for updating and inserting into the databank. Also set up hosting on Heroku.



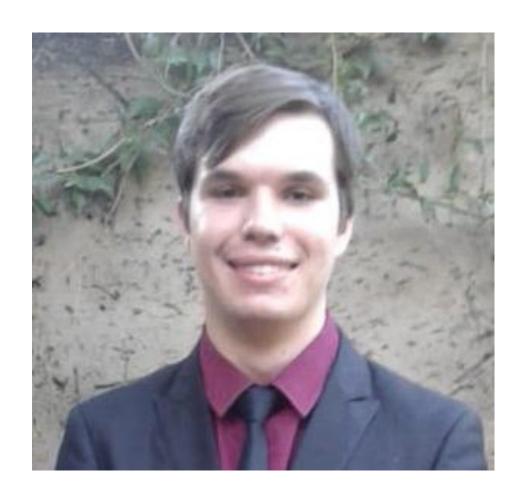
KOKETSO MOLAWA



Responsible for integrating the facial recognition library provided by node.js, into the overall subsystem. This involves writing methods to also loop through the entire JSON databank for do the facial recognition and return the ClientID.

JUSTIN ROSS

Responsible for writing the GET function responsible for interfacing with the databank module as well as providing the Authentication Subsystem with and interface to access the Facial Recognition. Also responsible for creating the LOGGING subsystem within the facial recognition subsystem.



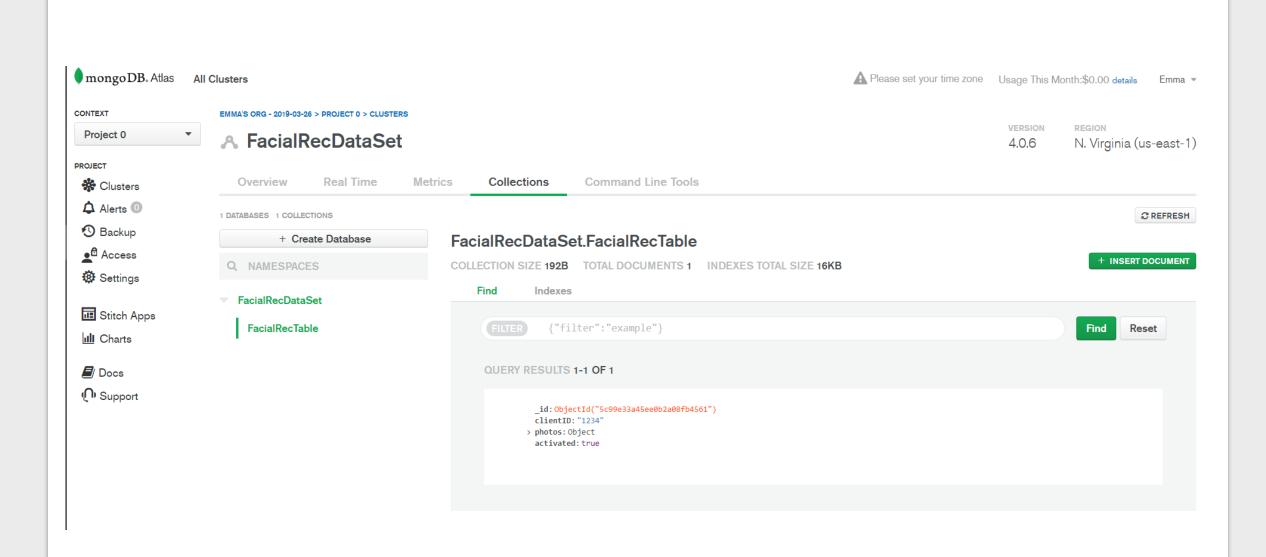
LIVE HOSTING

HEROKU

FOR THE DATABASE, API, AND LOG

UNIT TESTING

```
\blacktriangleleft
                               database.js
     const MongoClient = require('mongodb').MongoClient;
     const uri = "mongodb+srv://admin:admin123@facialrecdataset-umcor.mongodb.net/facialrecdataset?ssl=true&authSource=admin";
     const client = new MongoClient(uri, { useNewUrlParser: true });
     exports.Insert = function(client_id, images_json){
         client.connect(err => {
             if(err) throw err;
             const collection = client.db("FacialRecDataSet").collection("FacialRecTable");
             var obj = {clientID : client_id, photos : images_json, activated : true};
10
             collection.insertOne(obj, function(error, result){
11
                 if(error) throw error;
12
                 console.log("Inserted");
13
                 else return client id;
             });
             client.close();
17
         });
     };
```



Additional DB Testing File:

https://github.com/u17079544/Merlot-Facial-Recognition/blob/master/test/database/database-test.js



```
Update(client_id, images)

| updates the record in the database

Delete(client_id)
| deactivates the record in the database
```

```
var request = require('request');
var expect = require('chai').expect;
describe('post()', function(){
   it('should authenticate an image and return a clientID', function(){
       var sent={
              type: "authenticate",
              var expected={clientid: 1234};
                                                                  post()
       var result;
       request.post(
                                                                    should authenticate an image and return a clientID
       'http://localhost:3000/',
       { json: sent },
                                                                     should return an error
      function (error, response, body) {
          if (!error && response.statusCode == 200) {
                                                                    should return an error
              result=body;
              expect(result).to.be.equal(expected);
                                                                  4 passing (77ms)
      );
   });
   it('should add an array of images and return a boolean for success', function(){
       var sent={
              type: "update",
              images:
                  "/9j/4AAQSkZJRgABAgEAYABgAAD/7gAOQWRvYmUAZAAAAAAB/+EUI0V4aWYAAE1NACoAAAAIAAcBMgACAAAAFAAAAGIBOwACAAAABwAAAHZHRgADAAAAAQAEAABHSQADAAAAAQA/AACcnQABAAAAADgAAAAADqHAAHAAA
                 "/9j/4AAQSkZJRgABAgEAYABgAAD/7gAOQWRvYmUAZAAAAAB/+EUI0V4aWYAAE1NACoAAAAIAAcBMgACAAAAGIBOwACAAAABwAAAHZHRgADAAAAAQAEAABHSQADAAAAAQA/AACcnQABAAAAADgAAAAADqHAAHAAA
                 "/9j/4AAQSkZJRgABAgEAYABgAAD/7gAOQWRvYmUAZAAAAAB/+EUI0V4aWYAAE1NACoAAAAIAAcBMgACAAAAFAAAAGIBOwACAAAABwAAAHZHRgADAAAAAQAEAABHSQADAAAAAQA/AACcnQABAAAAADgAAAAADqHAAHAAA
                 "/9j/4AAQSkZJRgABAgEAYABgAAD/7gAOQWRvYmUAZAAAAAB/+EUI0V4aWYAAE1NACoAAAAIAAcBMgACAAAAFAAAAGIBOwACAAAABwAAAHZHRgADAAAAAQAEAABHSQADAAAAAQA/AACcnQABAAAAADgAAAAADqHAAHAAA
                  "/9j/4AAQSkZJRgABAgEAYABgAAD/7gAOQWRvYmUAZAAAAAAB/+EUI0V4aWYAAE1NACoAAAAIAAcBMgACAAAAFAAAAGIBOwACAAAABwAAAHZHRgADAAAAAQAEAABHSQADAAAAAQA/AACcnQABAAAAADgAAAAADqHAAHAAA
       var expected={success: true};
```

Link to Log Test: https://github.com/u17079544/Merlot-Facial-Recognition/blob/master/test/log/LogTest.js

```
add
   Should return a JSON array containing the added element (568ms)
   should return an error
   should return an error
get
  ¶ should return a JSON array containing a set of elements within the specified
oundaries (524ms)
  should return an error
8 passing (1s)
```

LINKS TO THE TECH THAT WE USED

GITHUB, ZENHUB AND LANDING PAGE







PROJECT MANAGEMENT TOOL:

HTTPS://GITHUB.COM/U17079544/MERLOT-FACIAL-RECOGNITION/BLOB/MASTER/VIEWS/PAGES/INDEX. EJS#ZENHUB **LANDING PAGE:**

HTTPS://CS.UP.AC.ZA/TEAMS/PAGES/SITE_VIEW/36

GITHUB RESPOSITIORY:

HTTPS://GITHUB.COM/U17079544/MERLOT-FACIAL-RECOGNITION