CS 301

Project by:

Aryansh Kurmi- 2018EEB1141
Suvansh Bhargava- 2018ceb1032

For this project, we mainly used the **Flask framework** to develop our portal website and we stored all the information in MongoDB.

Since our project was based on Flask Framework, we used Mongoclient in Pymongo to connect the MongoDB database with our project.

This code snippet shows the faculty Register method that we used to get the login credentials of the faculty and then we stored it into our database’s user collection.

```
@app.route('/register', methods=['POST', 'GET'])
def register():
    if request.method == 'POST':
        users = mongo.db.users
        existing_user = users.find_one({'name' :
request.form['username']})

        if existing_user is None:
            hashpass =
bcrypt.hashpw(request.form['password'].encode('utf8'), bcrypt.gensalt())
            users.insert({'name':request.form['username'], 'password':
hashpass})
            session['username'] =  request.form['username']
            session['logged_in'] = True
            return render_template('save_info.html')
        return 'That username already exists!'
    return render_template('register.html')
```

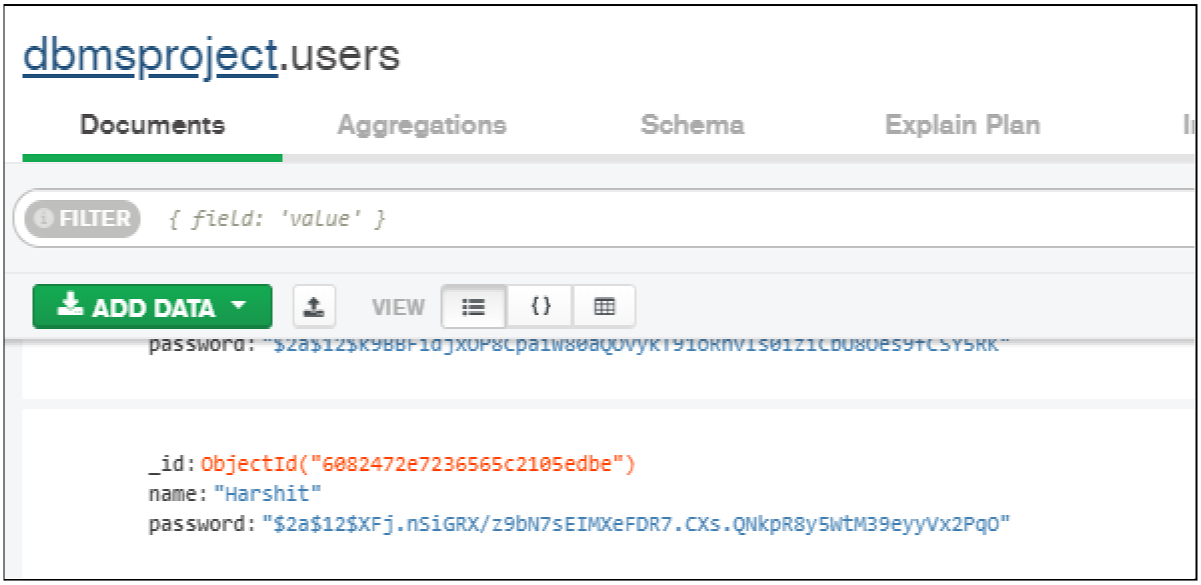
This code snippet shows the code where we updated all the information of the faculty. We used the update faculty information into the database’s profile collection.

```
@app.route('/updateinfo',methods=['POST', 'GET'])
def updateinfo():
  if request.method == 'POST':
    profile = mongo.db.profile
    form_data = request.form
    research_areas = []
    course = []
    publication_title = []
    publication_desc = []
    for key, values in form_data.items():
      if key.find("research") != -1:
        research_areas.append(values)
      elif key.find("course") != -1:
        course.append(values)
      elif key.find("title") != -1:
        publication_title.append(values)
      elif key.find("description") != -1:
        publication_desc.append(values)

profile.update({'username':session['username']},{"$set":{'username':sessio
n['username'],
                    'name':request.form['name'],
                    'email':request.form['email'],
                    'background':request.form['background'],
                    'department':request.form['department'],
                    'designation':request.form['designation'],
                    'course':course,
                    'research':research_areas,
                    'title':publication_title,
                    'description':publication_desc}})
    connect = db.connect()
    cursor = connect.cursor()
    cursor.execute("UPDATE Faculty SET name = %s, designation = %s,
department = %s WHERE username = %s",(request.form['name'],
request.form['designation'], request.form['department'],
session['username']))
    connect.commit()
    cursor.close()
    if(request.form['designation'] == 'Faculty'):
      return userprofile()
    return adminprofile()
```

After this, **for the leave application, we used SQL queries**(*we have sent you our queries and ER diagrams during phase A)*.

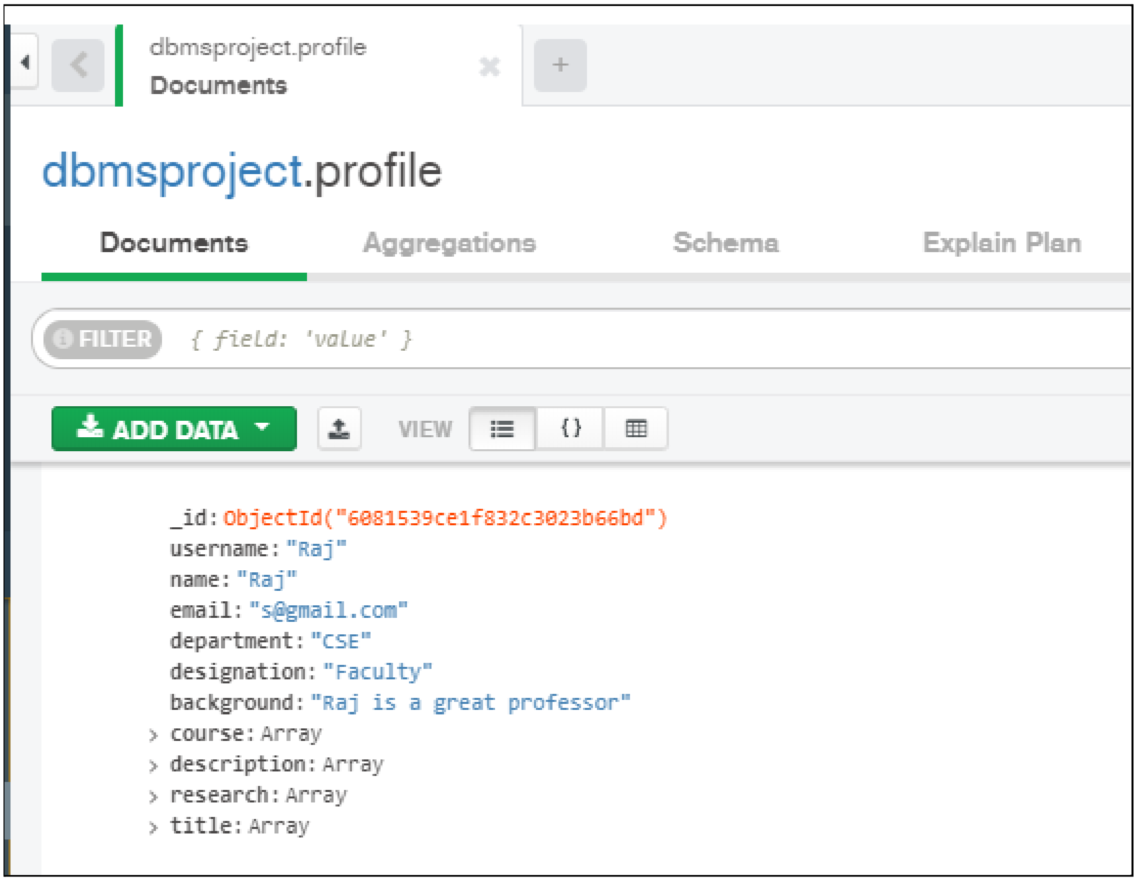
In MongoDB, we **created a database** named **dbmsproject** and in that database, we **created 2 collections** namely **users** and **profile** where we stored the login credentials and user information respectively. The user collection consisted of the username and password and the profile section had user information such as their username, name, email Id, department, designation, background, course, description, and research.



```
password: "$2a$12$k9BBF1djXOP8Cpa1W80aQOvyKT91oKnv1s01Z1CbU8Oes9fCSY5Rk"

_id: ObjectId("6082472e7236565c2105edbe")
name: "Harshit"
password: "$2a$12$XFj.nSiGRX/z9bN7sEIMXeFDR7.CXs.QNkpR8y5WtM39eyyVx2PqO"
```

User collection



```
 _id: ObjectId("6081539ce1f832c3023b66bd")
 username: "Raj"
 name: "Raj"
 email: "s@gmail.com"
 department: "CSE"
 designation: "Faculty"
 background: "Raj is a great professor"
> course: Array
> description: Array
> research: Array
> title: Array
```

Profile collection