**tutorial\_routes.py**

@app.route('/api/tutorial/<username>/<tutorial\_uuid>', methods=['GET'])

def get\_one\_tutorial(username, tutorial\_uuid):

    sql\_query = "SELECT \* FROM diyup.tutorials WHERE author\_username=%s AND uuid=%s"

    cur = mysql.connection.cursor()

    cur.execute(sql\_query, (username, tutorial\_uuid))

    tutorial = cur.fetchone()

    if not tutorial:

        return jsonify({'message' : 'No tutorial found!'}), 400

    sql\_query = "SELECT \* FROM diyup.steps WHERE tutorial\_uuid=%s"

    cur.execute(sql\_query, (tutorial[0],))

    steps = cur.fetchall()

    output\_steps = []

    tutorial\_data = {}

    tutorial\_data['uuid'] = tutorial[0]

    tutorial\_data['author\_username'] = tutorial[1]

    tutorial\_data['title'] = tutorial[2]

    tutorial\_data['image'] = tutorial[3]

    tutorial\_data['category'] = tutorial[4]

    tutorial\_data['description'] = tutorial[5]

    tutorial\_data['author\_difficulty'] = str(tutorial[6])

    tutorial\_data['viewer\_difficulty'] = str(average\_rating\_type\_for\_tutorial('difficulty', tutorial[0]))

    tutorial\_data['rating'] = str(average\_rating\_type\_for\_tutorial('score', tutorial[0]))

    for step in steps:

        step\_data = {}

        step\_data['index'] = step[1]

        step\_data['content'] = step[2]

        step\_data['image'] = step[3]

        output\_steps.append(step\_data)

    tutorial\_data['steps'] = output\_steps

    cur.close()

    return jsonify({'tutorial' : tutorial\_data}), 200

**user\_routes.py**

@app.route('/api/login', methods=['POST'])

def login():

    data = request.get\_json()

    username = data['username']

    password = data['password']

    if not username:

        return make\_response('Could not verify auth', 401, {'WWW-Authenticate' : 'Basic realm="Login required!"'})

    sql\_query = "SELECT \* FROM diyup.users WHERE username=%s"

    cur = mysql.connection.cursor()

    cur.execute(sql\_query, (username,))

    user = cur.fetchone()

    cur.close()

    if not user:

        return make\_response('Could not verify user', 401, {'WWW-Authenticate' : 'Basic realm="Login required!"'})

    if check\_password\_hash(user[2], password):

        token = jwt.encode({'email\_address' : user[0]}, app.config['SECRET\_KEY'])

        return jsonify({'token' : token.decode('UTF-8')})

    return make\_response('Could not verify', 401, {'WWW-Authenticate' : 'Basic realm="Login required!"'})

**comment\_routes.py**

@app.route('/api/comments/<tutorial\_uuid>/create/<reply\_comment\_id>', methods=['POST'])

@token\_required

def reply\_to\_tutorial\_comment(current\_user, tutorial\_uuid, reply\_comment\_id):

    data = request.get\_json()

    cur = mysql.connection.cursor()

    cur.execute("SELECT \* FROM diyup.comments WHERE tutorial\_uuid=%s AND reply\_to=%s", (tutorial\_uuid, reply\_comment\_id,))

    uuid = cur.fetchall()

    if not uuid:

        return jsonify({'message' : 'No tutorial ID found!'}), 400

    cur.execute("SELECT \* FROM diyup.comments ORDER BY id DESC LIMIT 1")

    index = cur.fetchone()

    id = int(index[0]) + 1

    content = data['content']

    image = data['image']

    reply\_to = reply\_comment\_id

    edited = 0

    date = time.ctime(1574039538)

    timestamp = date

    cur.execute("INSERT INTO diyup.comments(comments.tutorial\_uuid, username, content, created, timestamp, edited, image, reply\_to) VALUES(%s, %s, %s, %s, %s, %s, %s, %s)",

                (tutorial\_uuid, current\_user[1], content, date, timestamp, edited, image, reply\_to,))

    mysql.connection.commit()

    cur.close()

    return jsonify({'message' : 'Reply created!', 'comment id' : id}), 201