SW Engineering CSC648/848 Spring 2022

GatorJobs

Section 02 Team 05

Jasneil Nat - Team Lead

jnat1@mail.sfsu.edu

Nomar Olivas -Back End Lead

nolivas@mail.sfsu.edu

Risheek Agrawal - Front End Team

ragrawal1@mail.sfsu.edu

Purva Zinjarde - Front End Lead

pzinjarde@sfsu.edu

Shem Cheng - GitHub Lead

scheng8@mail.sfsu.edu

Yasin Hagos - Back End Team

yhagos@mail.sfsu.edu

Milestone 4
19 April 2022

Date Submitted:	**19 April 2022**
Date Revised:	**25 April 2022**

1. Product Summary

Name: GatorJobs

System:

(P1) System shall have a homepage.
(P1) System shall have admin pages.
(P1) System shall have an admin login page.
(P1) System shall have a company login page.
(P1) System shall have a student login page.
(P1) System shall have a student page.
(P1) System shall have a company page.
(P1) System shall give alerts to students if any jobs are available.
(P1) System shall have a search bar / filters.

Admin:

(P1) Admin shall be able to view and filter students and company data.
(P1) Admin shall be able to notify students about available job offers.
(P1) Admin shall be able to filter jobs.
(P1) Admin shall be able to filter and match students.

Student:

(P1) Students shall be able to create and register accounts using their sfsu email or ID number.
(P1) Students shall receive job alerts if some job description matches their portfolio.
(P1) Students shall be able to search and filter for tech jobs using the search bar or filter tool at the top of the page.
(P1) Students shall be able to add a link to their websites or projects. Links can be student's own personal websites or GitHub page
(P2) Students shall be able to upload a picture of themselves to their profile.
(P3) Students shall be able to upload recommendations from professors.

Companies:

(P1) Companies shall be able to create and register accounts.
(P1) Companies shall be able to post job availability by posting about the job titles, descriptions and skill set requirements.
(P2) Companies shall be able to upload a company picture or logo to their profile.

Unique: Ability to ask professors for a recommendation

URL to Product: 34.102.11.81/home

2. Usability Test Plan

Testing Function: Admin Matching Student with Jobs

Testing Objective: Ensure administrative users can match prospective students to an appropriate job.

Test Background and Setup: The system will be set up in the Google Cloud Platform using a virtual machine. A focus group will be assembled to help test the designated function.

The administrator is the intended user which will be able to connect the students and companies.

Usability Task Description:

Effectiveness would be measured by the question “It was easy to connect the Student to appropriate Jobs”. An average between 4 and 5 would signal that the GatorJobs website has a streamlined process for making sure students are finding appropriate job companies.

Efficiency can be measured through all questions as users complete the task of matching a student to jobs. If their responses are favorable to the interface being easy to use, GatorJobs will be found to be efficient.

Likert Scale Criteria:

- 1 = Strongly disagree
- 2 = Disagree
- 3 = Neutral
- 4 = Agree
- 5 = Strongly agree

Likert Questions to test User Satisfaction:

I found it easy to log in.

The user interface was simple to use.

I like the design of GatorJobs.

It was easy to connect the Student to appropriate Jobs.

3. QA Test Plan:

Test #	Title	Desc.	Input	Expected	Result*
1	Matching Students with Jobs	Attempt to match student with job according to specified filters in description	Student ID and job ID	Boolean (True if match, False if not match)	Currently in progress. Unable to process incomplete results
2	Login	Test login capabilities by entering existing username and password	Username and password	Boolean (True if login success, False if login failure)	Able to get boolean back but user able to login regardless of inputs
3	Search, Filter	Test search and filtering of students and jobs	Domain, skills, educational level/job title	List of students/jobs that match the criteria	Functioning as desired

***Tested on 4 Browsers**

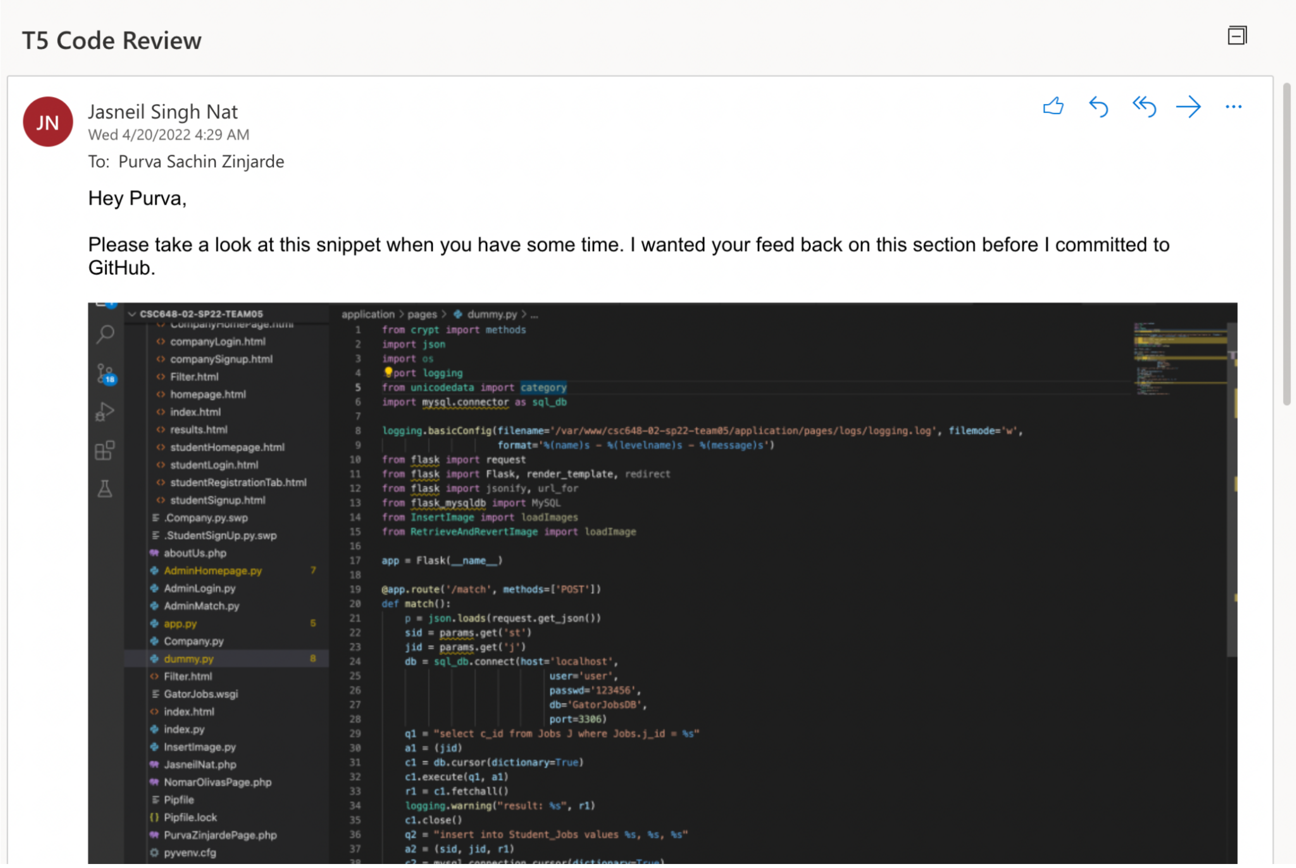
- **Google Chrome**
- **Opera**
- **Microsoft Edge**
- **Safari**

4. Code Review:

The coding style for Python and JavaScript have been used for their respective pages and modules. Indentation, white spaces, capitalization, style and spelling, use & style of comments is as per the unique coding style of each language.

A specific convention of variable names, function names, file organization, and comments has been followed throughout the project.

We have made it a point to peer review every module or page so that there is uniformity throughout the application. Here is an example of peer code review on admin matching students with the jobs.



T5 Code Review

Jasneil Singh Nat
Wed 4/20/2022 4:29 AM
To: Purva Sachin Zinjarde

Hey Purva,

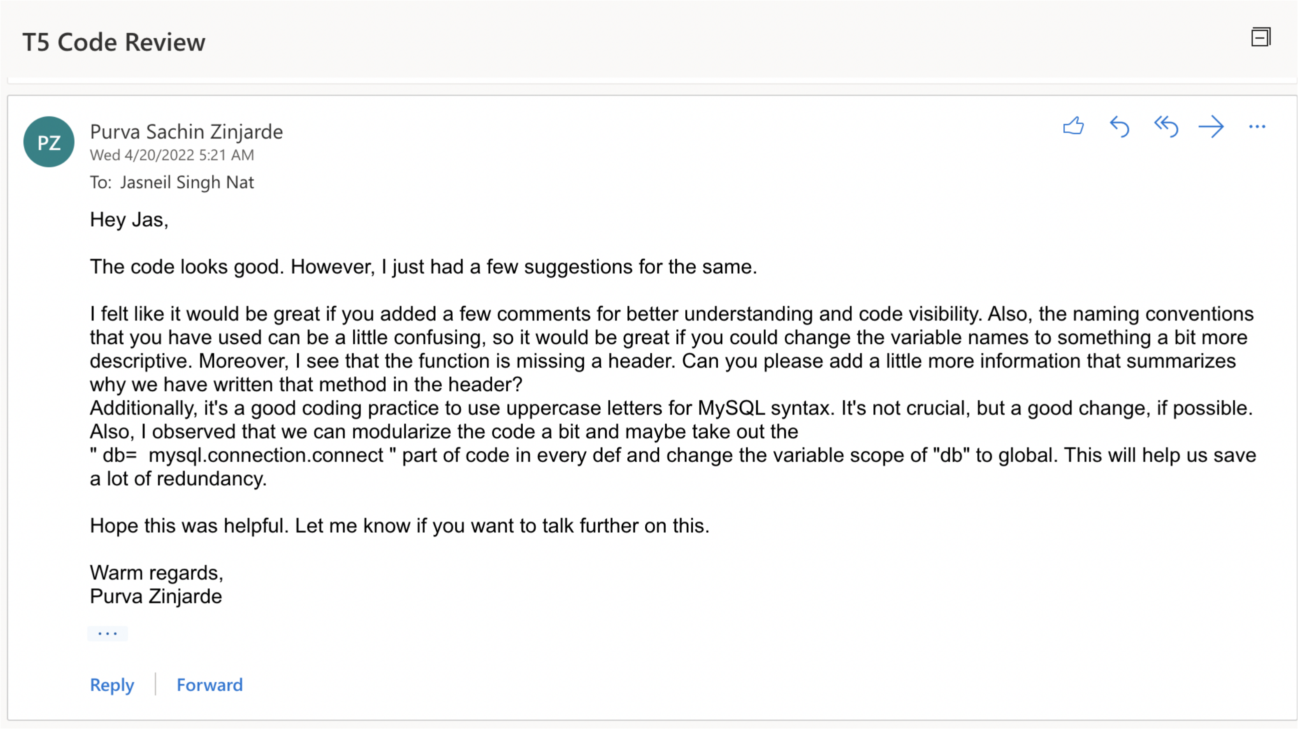
Please take a look at this snippet when you have some time. I wanted your feed back on this section before I committed to GitHub.

```python
from crypt import methods
import json
import os
💡port logging
from unicodedata import category
import mysql.connector as sql_db

logging.basicConfig(filename='/var/www/csc648-02-sp22-team05/application/pages/logs/logging.log', filemode='w',
                    format='%(name)s - %(levelname)s - %(message)s')
from flask import request
from flask import Flask, render_template, redirect
from flask import jsonify, url_for
from flask_mysqldb import MySQL
from InsertImage import loadImages
from RetrieveAndRevertImage import loadImage

app = Flask(__name__)

@app.route('/match', methods=['POST'])
def match():
    p = json.loads(request.get_json())
    sid = params.get('st')
    jid = params.get('j')
    db = sql_db.connect(host='localhost',
                        user='user',
                        passwd='123456',
                        db='GatorJobsDB',
                        port=3306)
    q1 = "select c_id from Jobs J where Jobs.j_id = %s"
    a1 = (jid)
    c1 = db.cursor(dictionary=True)
    c1.execute(q1, a1)
    r1 = c1.fetchall()
    logging.warning("result: %s", r1)
    c1.close()
    q2 = "insert into Student_Jobs values %s, %s, %s"
    a2 = (sid, jid, r1)
    c2 = mysql.connection.cursor(dictionary=True)
```



T5 Code Review

Purva Sachin Zinjarde
Wed 4/20/2022 5:21 AM
To: Jasneil Singh Nat

Hey Jas,

The code looks good. However, I just had a few suggestions for the same.

I felt like it would be great if you added a few comments for better understanding and code visibility. Also, the naming conventions that you have used can be a little confusing, so it would be great if you could change the variable names to something a bit more descriptive. Moreover, I see that the function is missing a header. Can you please add a little more information that summarizes why we have written that method in the header?
Additionally, it's a good coding practice to use uppercase letters for MySQL syntax. It's not crucial, but a good change, if possible.
Also, I observed that we can modularize the code a bit and maybe take out the
" db= mysql.connection.connect " part of code in every def and change the variable scope of "db" to global. This will help us save a lot of redundancy.

Hope this was helpful. Let me know if you want to talk further on this.

Warm regards,
Purva Zinjarde

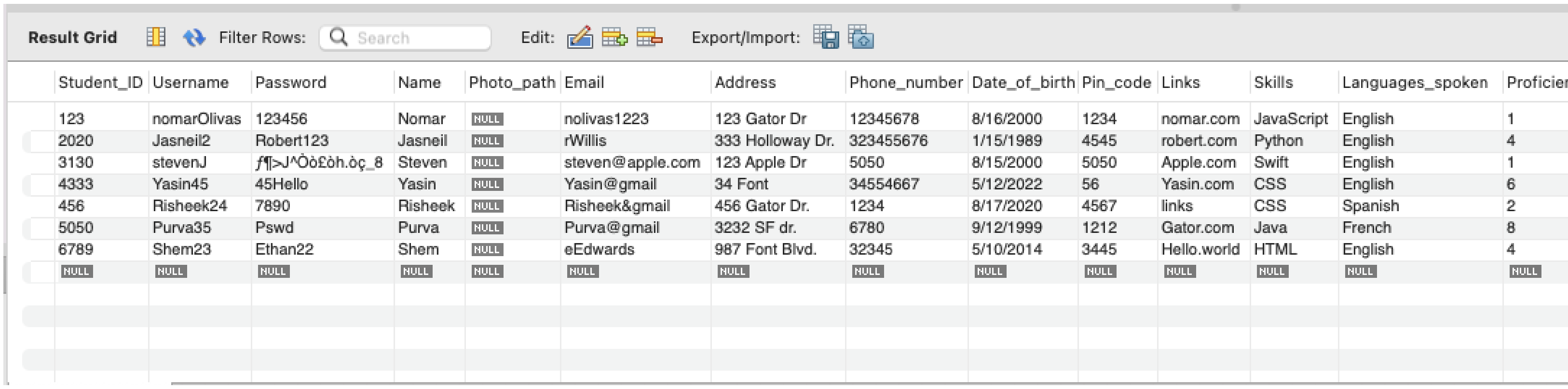
5. Self-Check on best practices for security:

Major Assets to be protected:

- Student's ID
- Student's Username
- Student's Password
- Student's Address
- Student's Phone number
- Student's Pin Code
- Student's Permissions
- Company's ID
- Company's Username
- Company's Password
- Company's email
- Company's Permissions

How are we protecting each asset:
For each major asset above, we are using MySQL's Enterprise Encryption which allows us to encrypt each asset using a public-key and a salt ID. In order to add the same password for different users, we first need to use a salt in order to decode the similar encrypted password.

Confirmed Encrypted password in the Database: In the picture below, line 3 shows an example of the encryption for the Student's password that is located in our GatorJobsDB server in MySQL. The same is done to the Company's password as well.



Student_ID	Username	Password	Name	Photo_path	Email	Address	Phone_number	Date_of_birth	Pin_code	Links	Skills	Languages_spoken	Proficier
123	nomarOlivas	123456	Nomar	NULL	nolivas1223	123 Gator Dr	12345678	8/16/2000	1234	nomar.com	JavaScript	English	1
2020	Jasneil2	Robert123	Jasneil	NULL	rWillis	333 Holloway Dr.	323455676	1/15/1989	4545	robert.com	Python	English	4
3130	stevenJ	ƒ¶>J^Òò£òh.òç_8	Steven	NULL	steven@apple.com	123 Apple Dr	5050	8/15/2000	5050	Apple.com	Swift	English	1
4333	Yasin45	45Hello	Yasin	NULL	Yasin@gmail	34 Font	34554667	5/12/2022	56	Yasin.com	CSS	English	6
456	Risheek24	7890	Risheek	NULL	Risheek&gmail	456 Gator Dr.	1234	8/17/2020	4567	links	CSS	Spanish	2
5050	Purva35	Pswd	Purva	NULL	Purva@gmail	3232 SF dr.	6780	9/12/1999	1212	Gator.com	Java	French	8
6789	Shem23	Ethan22	Shem	NULL	eEdwards	987 Font Blvd.	32345	5/10/2014	3445	Hello.world	HTML	English	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Confirm Input data validation: We can confirm that our compulsory fields that are inputted for Students and Companies which includes emails, passwords, phone numbers, dates, etc. are valid. Our search bar input is not more than 40 chars for filters.

6. Self-check: Adherence to original Non-functional specs

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO).

ON TRACK

2. Application shall be optimized for standard desktop/laptop browsers e.g., must render correctly on the two latest versions of two major browsers

DONE

3. Selected application functions must render well on mobile devices

DONE

4. Data shall be stored in the team's chosen database technology on the team's deployment server.

DONE

5. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.

ON TRACK - Validations are needed

6. The language used shall be English.

DONE

7. Application shall be very easy to use and intuitive.

DONE

8. Google maps and analytics shall be added

ON TRACK

9. No email clients shall be allowed. You shall use webmail.

ON TRACK

10. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.

DONE

11. Site security: basic best practices shall be applied (as covered in the class)

ON TRACK

12. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development

DONE

13. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only” at the top of the WWW page. (Important so not to confuse this with a real application).

DONE