ASSIGNMENT PART 1

Server-Side Programming

INDIVIDUAL REPORT

(Swift Outsource Company)

Student Name: Pui Wing Ng

Student ID: 103606281

COS10026 - COMPUTER TECHNOLOGY INQUIRY PROJECT

SWINBURNE UNIVERSITY OF TECHNOLOGY

Table of Contents

1. INTRODUCTION

The primary objective of this report is to provide a comprehensive analysis of the "Server-Side Programming" project. This further enhance the html website that was previously developed in Assignement1. It included a series of website functionality through server-side PHP, the creation of MySQL tables for data storage and management, and the development of a dedicated webpage for HR managers to interact with job applications. The objective of the report is to provide a comprehensive set of instructions and requirements for the assignment's Part 2, which involves further enhancing a website previously developed. Additionally, the report emphasizes the use of PHP for code modularity and reusability, the creation of a "settings.php" file for efficient database connection management, and the design of an "EOI" (Expressions of Interest) table with specified fields and statuses. The report also highlights the importance of server-side data validation and sanitization to ensure data integrity and user-friendly error handling. Overall, it serves as a guiding document for users to use this website, helping them understand the website's functionality and database integration effectively.

This report commences with an introduction detailing the report's objectives. This is followed by a comprehensive section on the website, which delves into website description, main functionalities and key features. A dedicated section addresses the security of the website, focusing on both researched web security improvements and their specific implementation within the EOI system. The subsequent section elucidates the personal contributions made to the project, while a reflective segment provides insights on

the professional experience gained and discusses pertinent web development issues like privacy or security. The report concludes with a summary of the discussed topics.

2. THE WEBSITE

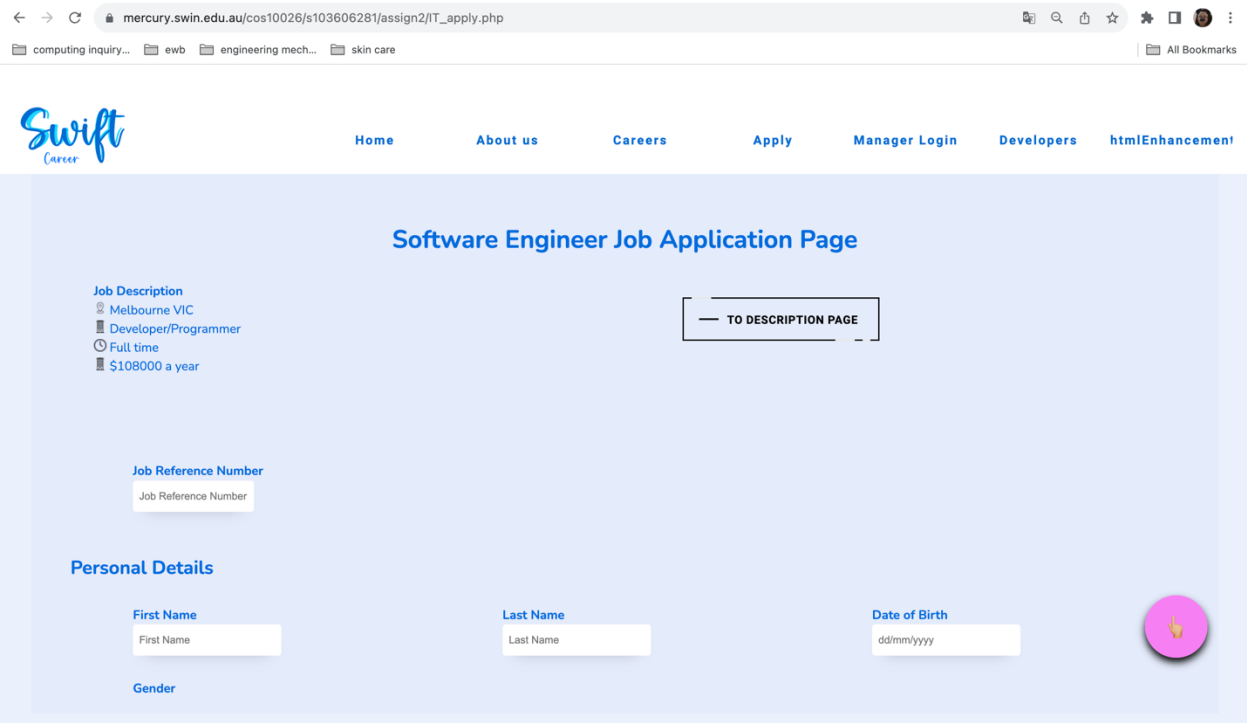

Figure 1: IT_apply.php

Figure 1 is IT_apply.php. There is an overview of the this software engineer job for us, there is job description button on the side for easy access. There is a right pink button on the right button to press, if user wants to go to the top of the page straight away.

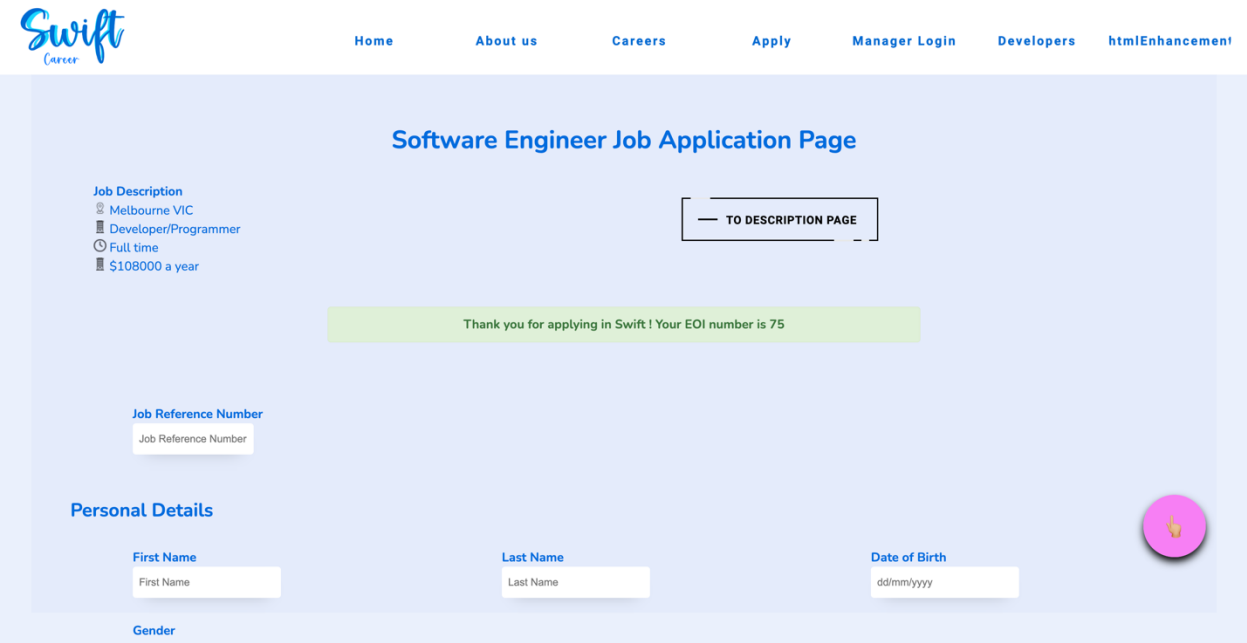

Figure 2: Successful submission message

Figure 2 describes that if applicants have successfully submitted the form, this success connfirmation message will pop out to applicants.

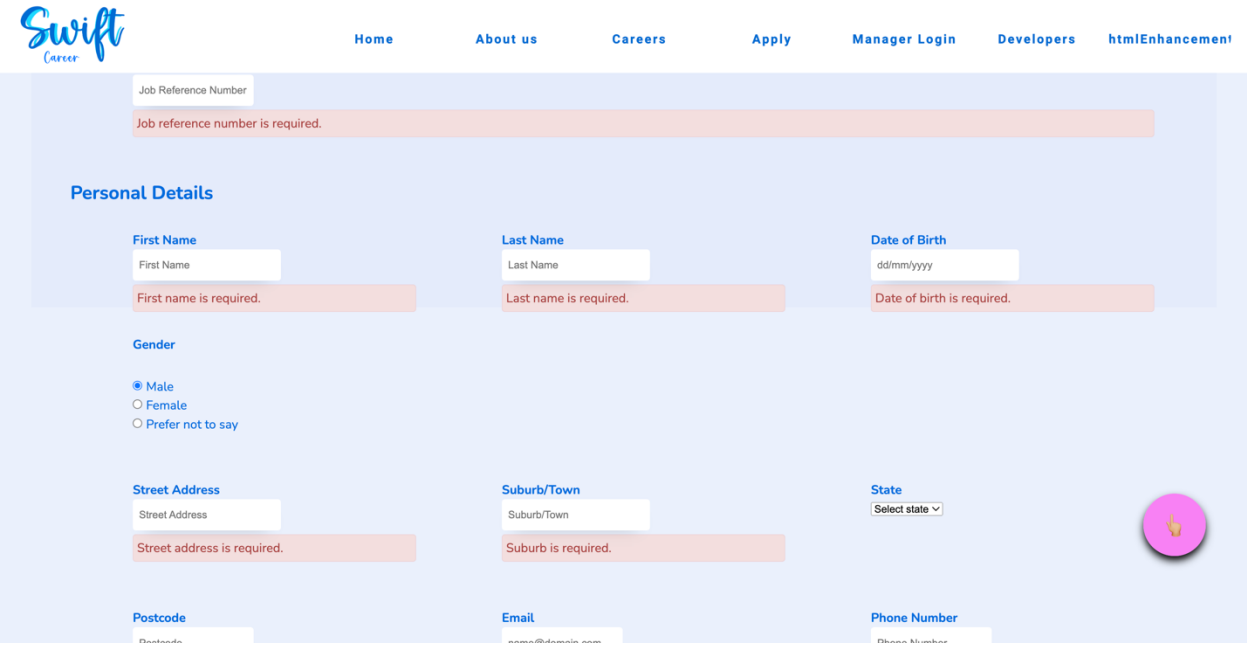

Figure 3: Server-side validation

Figure 3 explains that, if invalid information is entered in the application form, then the form will not submit, error messages will also pop up.

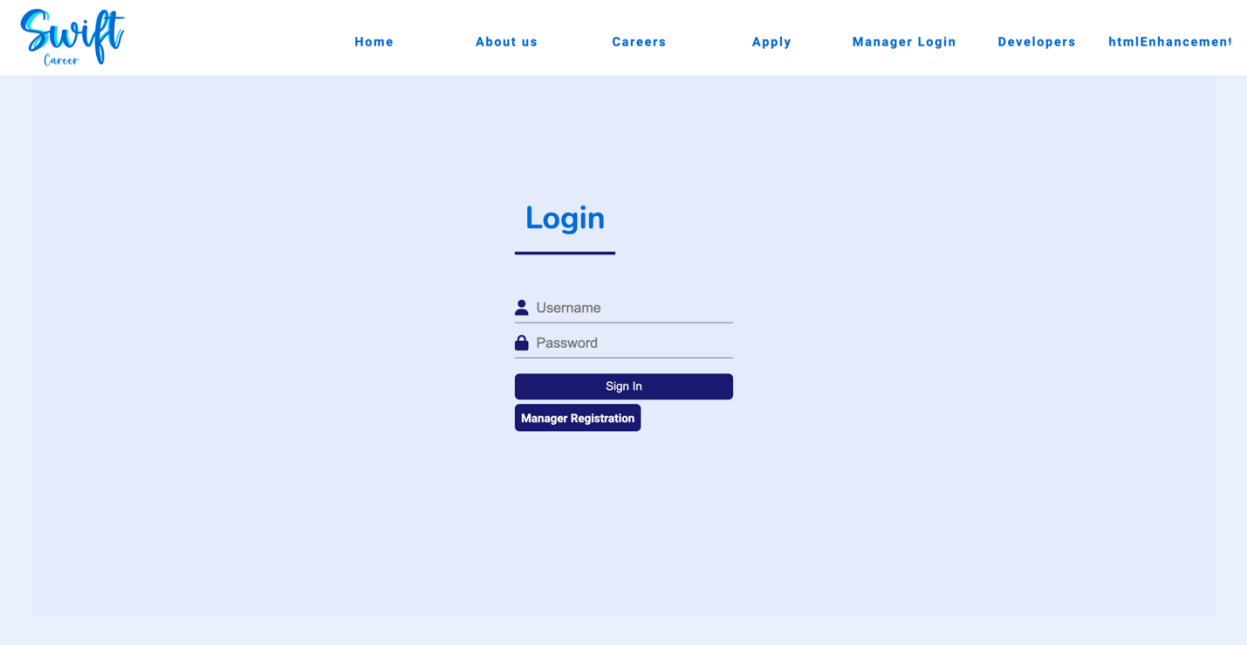

Figure 4: login1.php for manage.php

Figure 4 describes that manager will have to login to access manage.php. no one can access manage.php with logining in.

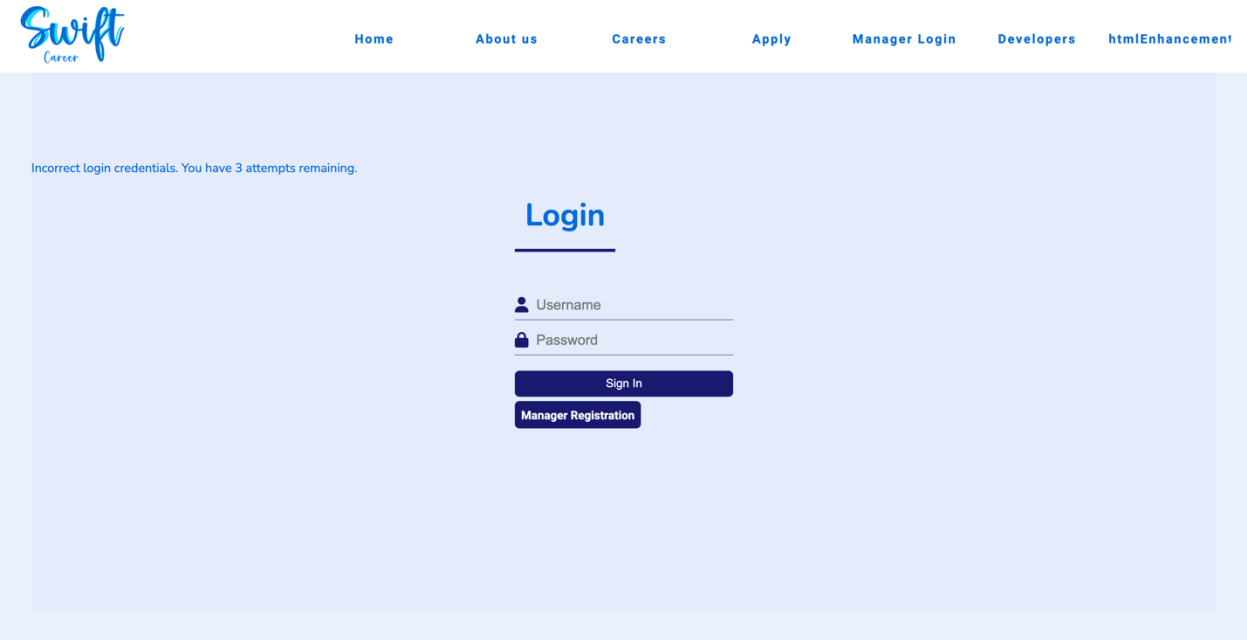

Figure 5: Invalid login attempts

Figure 5 shows that if invalid login information is entered, then there will be 3 attemps left for user to login.

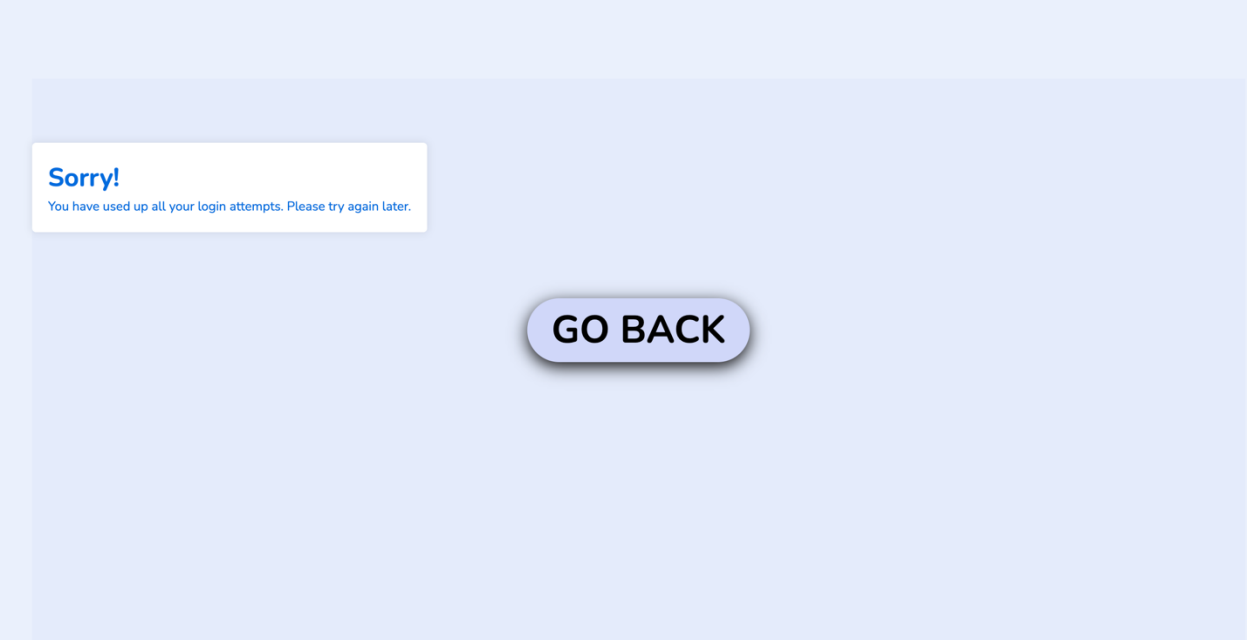

Figure 6: sorry.php

Figure 6 if all attempts used, then it will take user to sorry.php. And will ask the users to try again later.

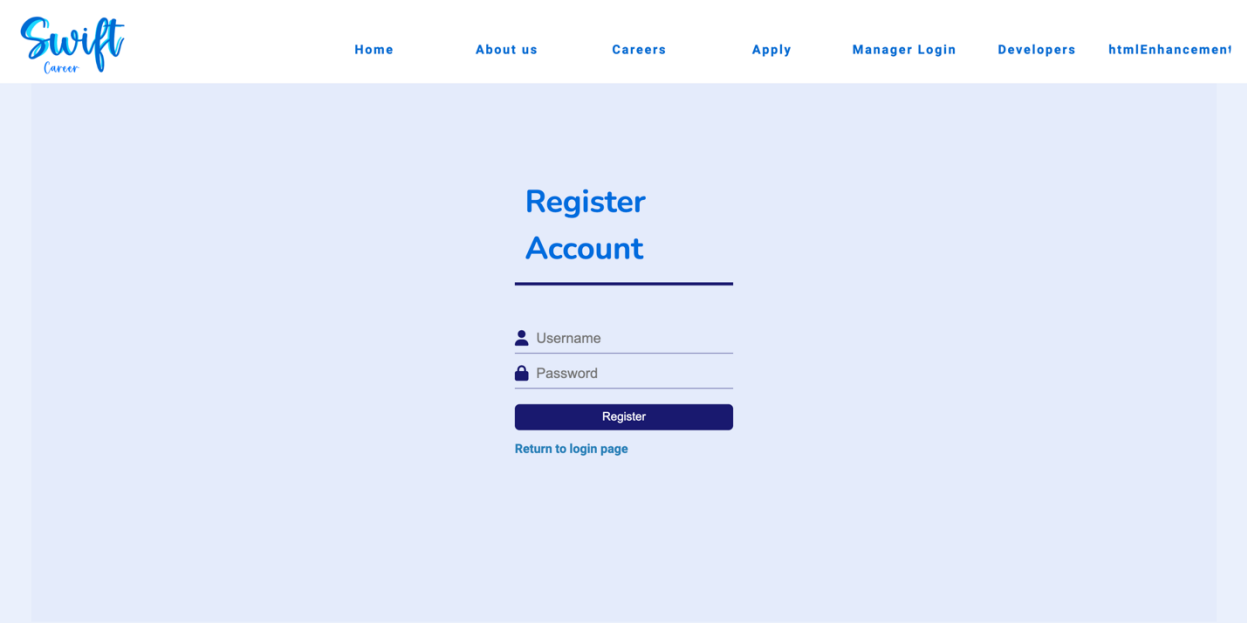

Figure 7: manager_registration.php

Figure 7 shows that if manager do not have an account, they can register an account here.

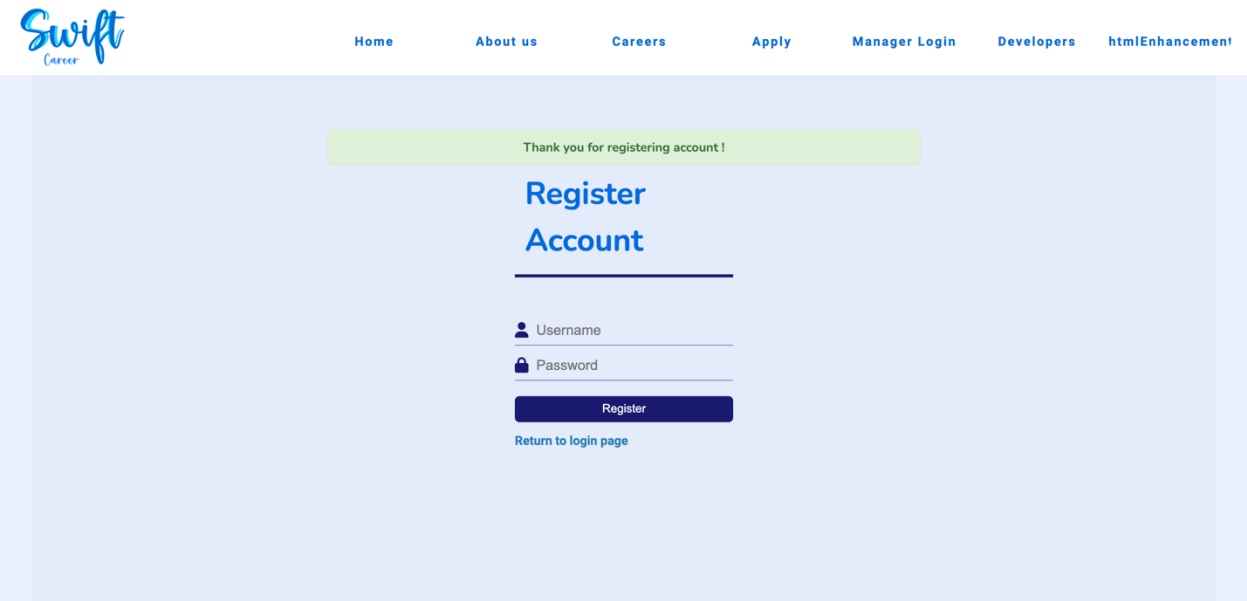

Figure 8: success message shows if registered successfully

Figure 8 if user registered an account successfully, a success message will pop out.

Figure 9: manage.php

Figure 9: A logout button and refresh table button, we can see the display table below. When you press the log out button, logout and destroy the session and take you back to login page.

Figure 10: the other side of the display table in manage.php

Figure 10 shows that all skills that was selected in application page is shown here, we can scroll the table to the other side to view more information.

Figure 11: show of status change in manage.php

In **Figure 11,** eoiNumber 64 changed from new to current. A message' status updated successfully' will pop up if the status if changed successfully.

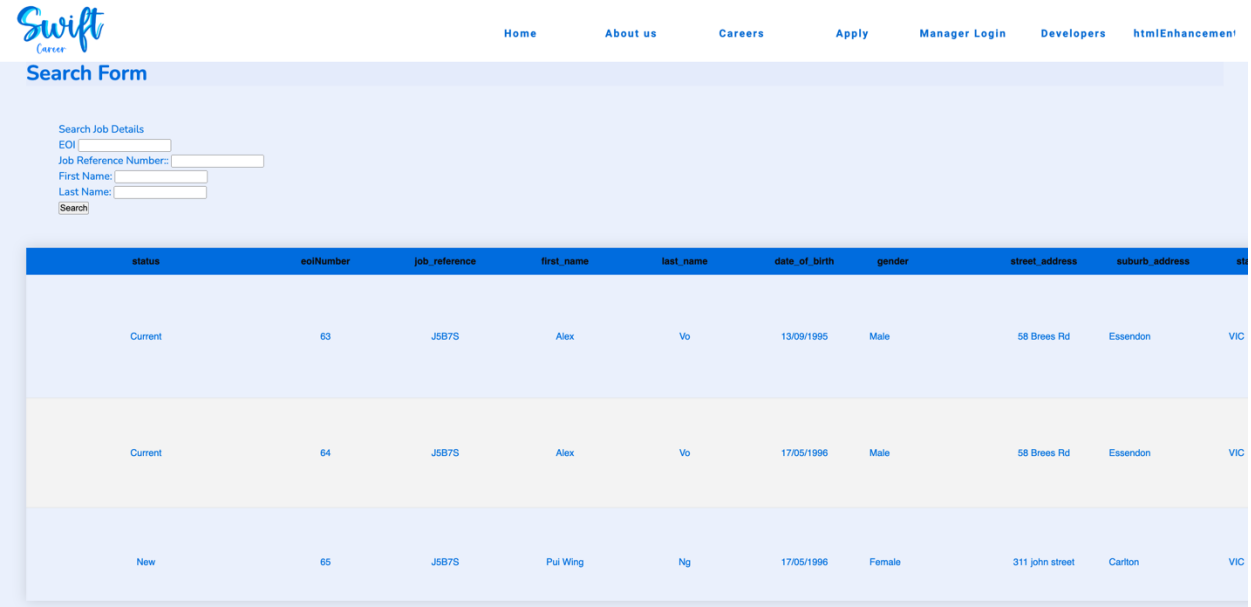

Figure 12: Search job reference number in manage.php

In **Figure 12,** search results when I search for job reference number J5B7S.

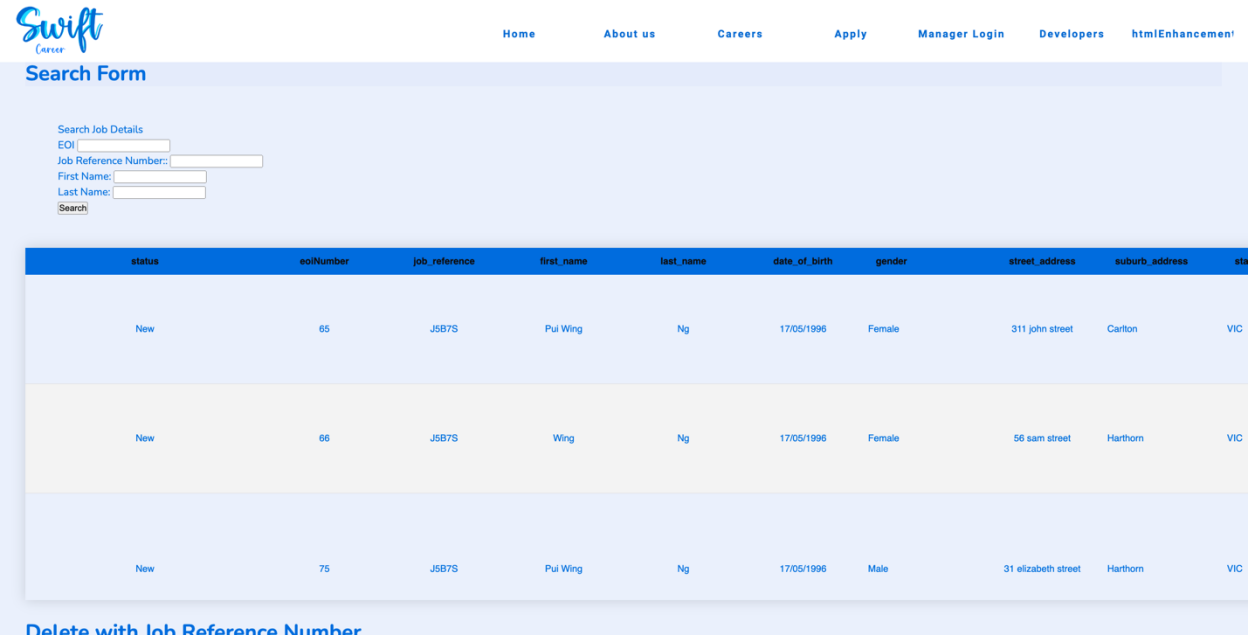

Figure 13: First name search results

In **figure 13**: when I type in ‘w’ or ‘wi’ in the first name, search results of Pui Wing, Wing will show.

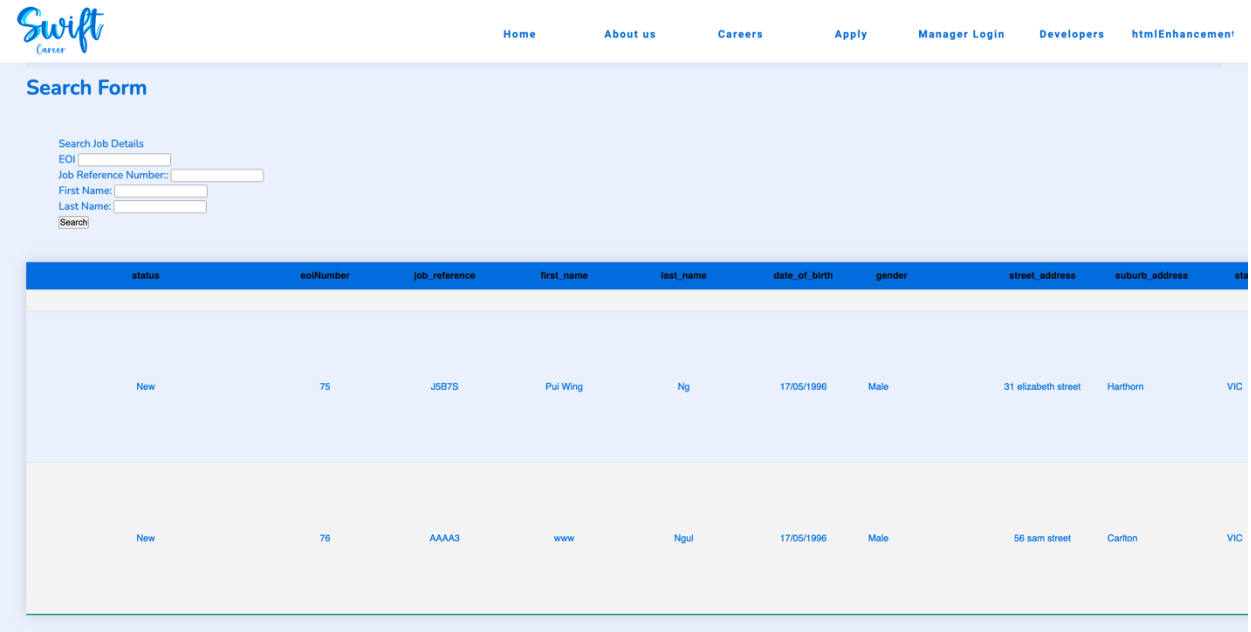

Figure 14: Last name search

Figure 14 shows the list of applicant's last name that contain the letter n after typing 'n' in last name to search.

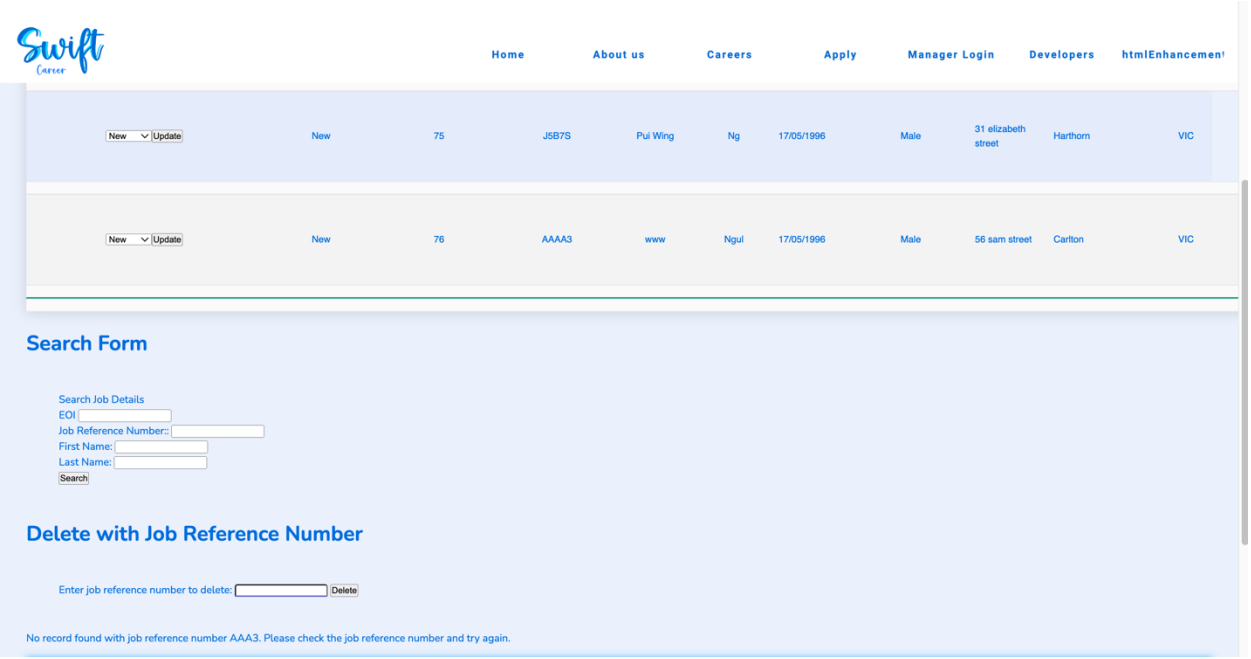

Figure 15: error message in the delete section if job reference number not found

In **figure 15**, manager can see a message 'No record found with job reference number AAA3. Please check the job reference number and try again' if they try to delete a job reference number that does not exist.

Figure 16: display table updated immediately when job reference number deleted

The display table on top is not showing the job reference number AAAA3 anymore. The table is updated immediately after deletion **(figure 16).**

Key Innovative Features:

1. Smooth Navigation: The button to return to the top of the page enhances user navigation.
2. Server-side Validation: This ensures data integrity and reduces the chances of erroneous or malicious data being stored.
3. Advanced Search Capabilities: Ability to search by job reference number, first name, and last name, providing managers with a flexible tool to filter applications.
4. Dynamic Content Rendering: Immediate updates on the table after operations like status change or deletion enhance the user experience.
5. Security Features: The account locking mechanism after multiple failed login attempts is a crucial feature to deter unauthorized access attempts.

In summary, the website is comprehensive and user-friendly for IT job application and application management. It uses a mix of server-side validation to provide a seamless experience for both applicants and managers, with a strong emphasis on data security and ease of navigation.

3. SECURITY OF THE WEBSITE

1. **Install Security Certificates and Use HTTPS**:

HTTPS stands for Hyper Text Transfer Protocol Secure. It's the secure version of HTTP, which is the protocol through which data is passed between a web browser and a website. HTTPS

uses SSL (Secure Socket Layer) to encrypt data transferred between a user's browser and the web server[1]. With HTTPS, data is encrypted in transit in both directions, ensuring that malicious parties can't observe or tamper with the data being sent. This means that usernames, passwords, and other sensitive information can't be stolen in transit when users enter them into a form. Especially for websites that handle sensitive data, using HTTPS is crucial to maintain user trust and protect their information (Krombholz et. al 2019). To implement HTTPS, website owners need to purchase and install an SSL certificate from a trusted Certificate Authority. Once the certificate is installed, the server should be configured to force HTTPS, ensuring all data is encrypted during transit[2].

2. **Hire a Security Expert**:

A security expert is a professional who specializes in identifying vulnerabilities in websites and suggesting measures to rectify them. Even with the best security practices in place, vulnerabilities can still exist. A security expert can conduct thorough penetration testing, identify weak points, and recommend solutions[3]. They can provide insights into the latest security threats and ensure that the website is fortified against them. Regular security audits should be scheduled. These audits can be a combination of automated scans and manual testing to ensure comprehensive coverage. Hiring a security expert provides an external perspective, which can be invaluable in identifying overlooked vulnerabilities[3].

3. **Implement Two-Factor Authentication (2FA)**:

2FA is an authentication method where users must provide two separate forms of identification before they can access their account[4]. Passwords alone are no longer considered sufficient for securing accounts. Even if a malicious actor obtains a user's password, they would still need the second form of identification to gain access. This makes unauthorized access significantly more challenging. There are various forms of 2FA, including SMS codes, authentication apps, and hardware tokens. Website owners should offer multiple 2FA options and encourage or even mandate its use, especially for accounts with elevated privileges[4].

4. **Enforce a Strong Password Policy**:

A strong password policy is a set of criteria that passwords must meet to be considered strong and secure from Windows[5]. The importance of such a policy cannot be overstated. Weak passwords can be easily guessed or cracked using brute force attacks, where attackers try numerous combinations until they find the right one. Additionally, common words or phrases, often found in a dictionary, can be used to guess passwords. A strong policy prevents the use of such easily guessable passwords, ensuring that user accounts remain uncompromised, which

[1] https://www.cloudflare.com/en-gb/learning/ssl/why-use-https/
[2] https://www.kaspersky.com/resource-center/definitions/what-is-a-ssl-certificate
[3] https://cybercx.com.au/resource/penetration-testing-guide/#what-is-pen-testing
[4] https://www.microsoft.com/en-au/security/business/security-101/what-is-two-factor-authentication-2fa#:~:text=2FA%20implementation,-Implementing%202FA%20within&text=These%20codes%20are%20sent%20via,an%20SMS%20code%20or%20token.
[5] https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/password-policy

in turn protects both the user's data and the website's reputation[5]. To implement a robust password policy, it's essential to mandate a mix of uppercase letters, lowercase letters, numbers, and special characters. Setting a minimum password length, often recommended to be at least 12 characters, further strengthens security. Users should be required to change their passwords periodically, but not too frequently to avoid user frustration. Implementing checks to prevent the use of commonly used passwords and encouraging the use of multi-word passphrases, which are longer and often more secure than typical passwords, can further bolster security. Where as in our manager_registration.php page, to register a password , the manager must provide a password that contains a upper case, lower case letter, and a number. Terrible passwords such as “123456” and “password” continue in the #1 and #2 spots[6], they are predictable, easily guessable passwords. Using these passwords will put anyone at substantial risk of being hacked and having their identities stolen.

5. **Use a Secure Host**:

Using a secure host means opting for a hosting provider that offers enhanced security features and prioritizes the protection of hosted websites. The foundation of a website's security starts with its hosting infrastructure. If the server is compromised, all hosted websites could be at risk. Secure hosts not only protect the data but also offer regular backups, ensuring that data can be restored in case of breaches or failures. Moreover, reliable hosts with good security measures ensure higher website availability, reducing the chances of downtime due to security incidents[7]. When choosing a host, it's essential to research their security features, customer reviews, and overall reputation in the industry. Features like Distributed denial-of-service (DDoS), robust firewalls to block malicious traffic, daily or real-time backups, and 24/7 support are crucial for a secure hosting experience[8].

All the above can be the improvements that is needed to be made for our Swift webpage

[6] https://teampassword.com/blog/worst-passwords-of-2018
[7] https://www.cloudways.com/blog/web-hosting-security/
[8] https://csguide.cs.princeton.edu/security/host

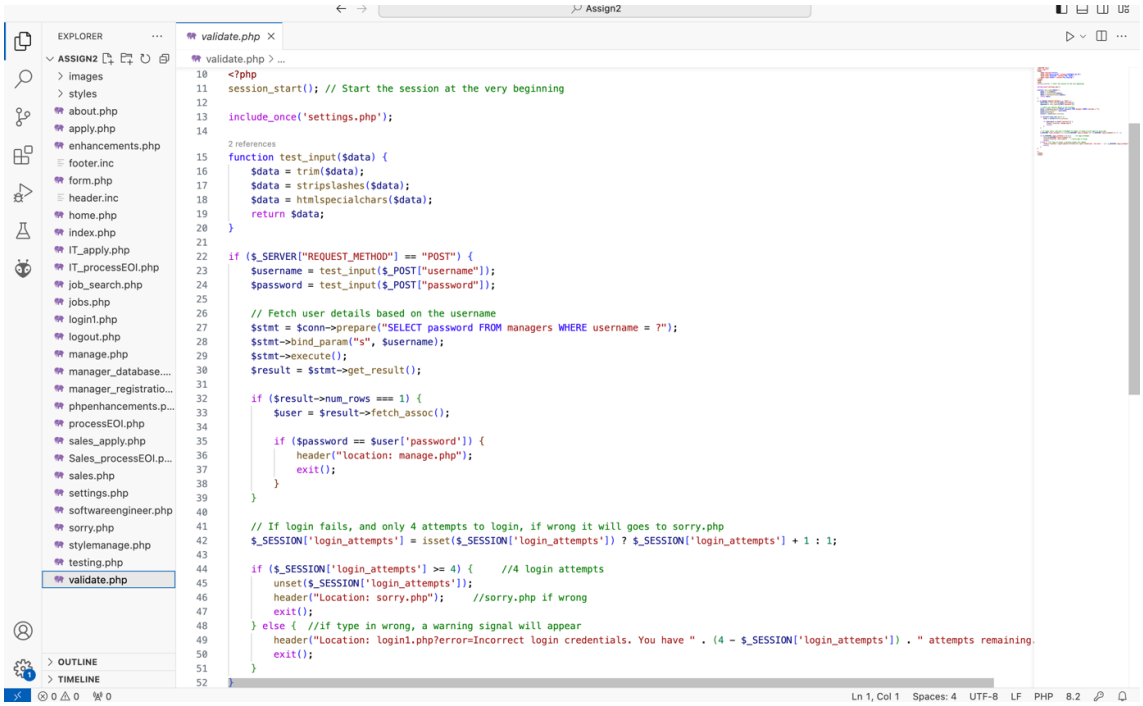

Figure 17:login1.php

Two-Factor Authentication (2FA) can be implemented here

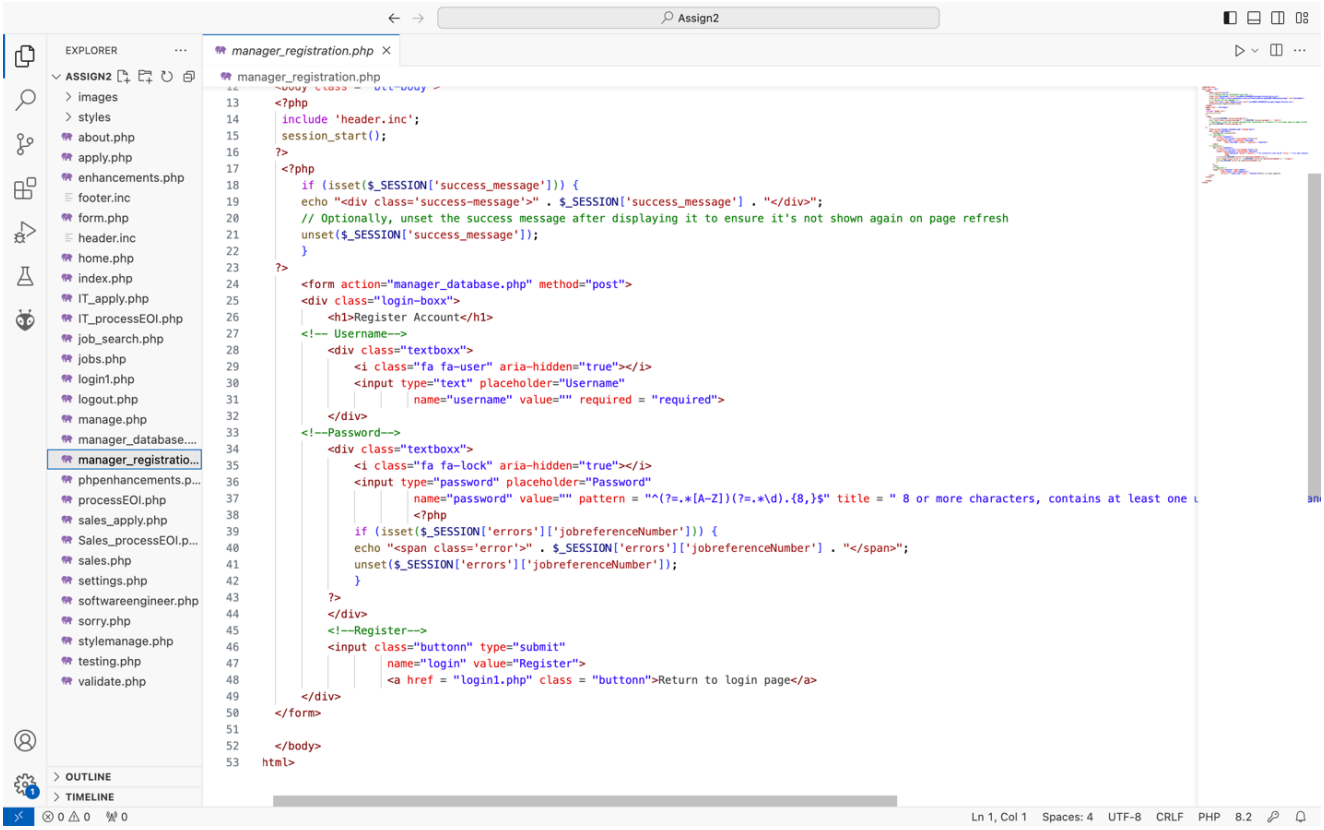

Figure 18: code of manager_registration.php

The strong password policy can be implemented here

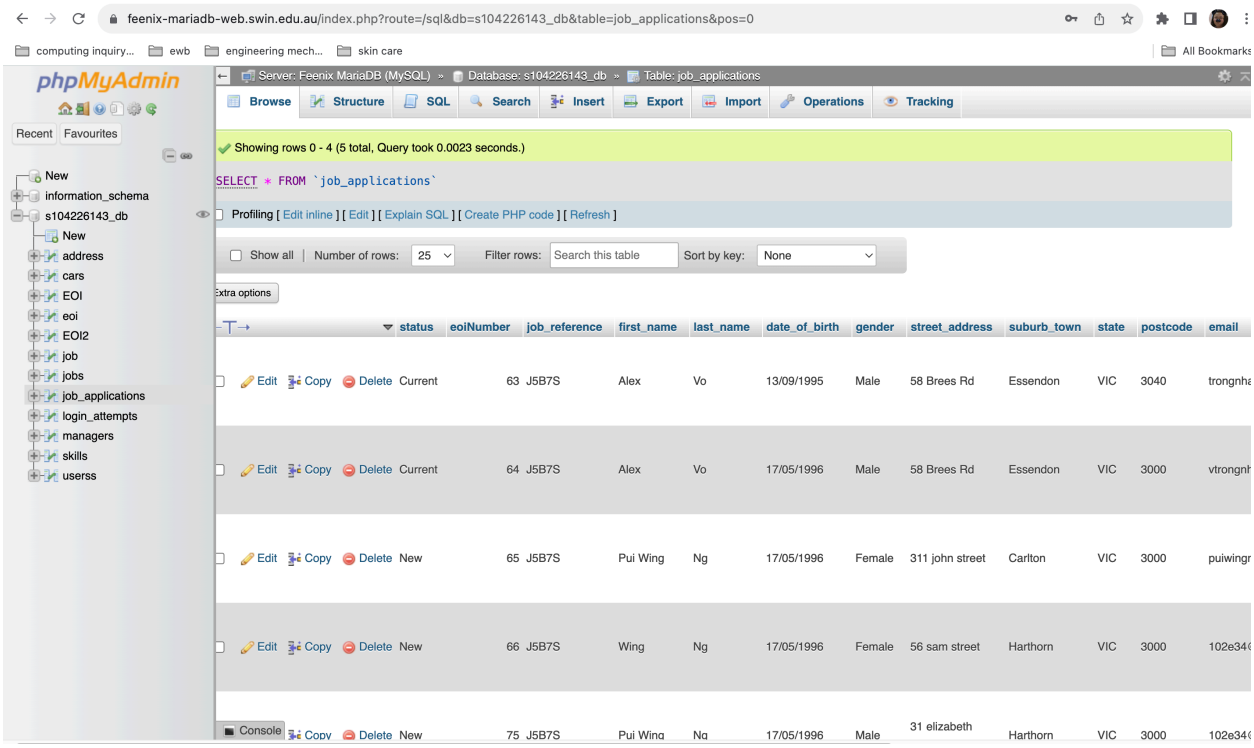

Figure 19:database

Secure host can be implemented here.

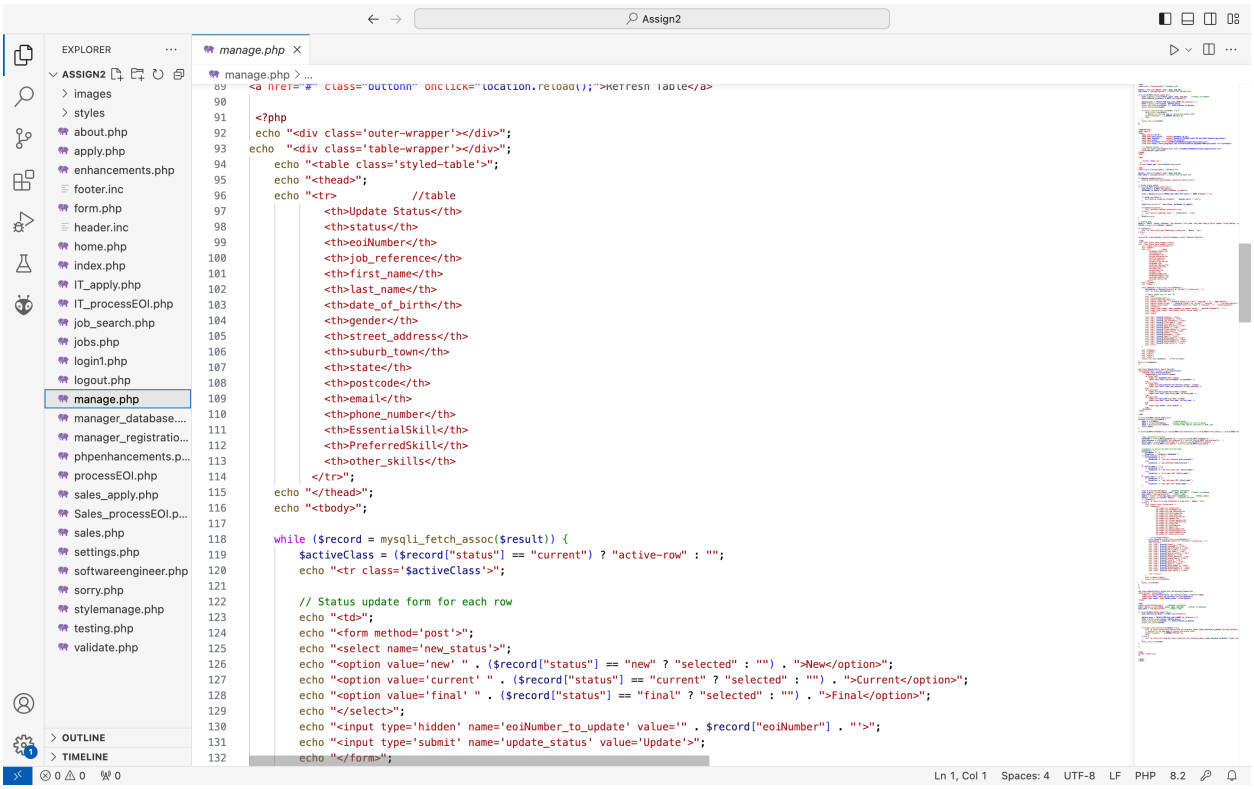

Figure 20:manage.php

Security certificates can be implemented here.

4. CONTRIBUTION

I am responsible for

- part 1: 'Use PHP to reuse common elements in your Website', everything other than the scrolling of table and logout button and logout.php. This includes, List all EOIs for a particular position (given a job reference number). List all EOIs for a particular applicant given their first name, last name or both. Delete all EOIs with a specified job reference number, Change the Status of an EOI
- part 5 'HR manager queries mange.php', and part 6 only login page and css in this group I am the group leader. I coordinate the whole group and open for questions when group mates have problems. I distribute them the work and remind them what need to be done. If they have problems with looking up of resources, I will help them with where to look up the resource.
- Partial of part 6 enhancement (login.php),and styling the page.

4.1. CHALLENGES

we only really actually started the assignment around 3weeks before the assignment is due. As we only started really understand want we need to do till then. But then we started super fast. And our group mate Alex did not have access to the mercury server since week 10, but we were not able to notice his challenges in time. We know this kinda in the last minute, and he did not reach out to teachers to So he will have to keep sending me his php, and check on my side. And challenges are we were so frustrated for a while that we are not able to get the table scrolling working in the manage.php. we have tried our best that the information in the table in all shown out, but not the table header. Also when we were working on the login.php, we were not able to figure out how to work out the disable login, if attempts all used. We suspect that might need java script to allow it to work. So in the end we used sorry.php if all attempts are being used. We got stuck on the processEOI.php for a while as well, when skills in the form is ticked and submitted, it did not show up in the database. Alex, me and Finn needed to all gather together and dicuss on how to work this out.

Where as for my part(manage.php). Taking reference from lab 10, I changed the looking of the table, so the page won't look that pale. But unfortunately when I changed the looking of the table, the table is so wide it is out of the webpage, even though we can still scroll it, this is not user friendly, we had no clue how to do, in the end we came to conclution that try to make the whole table able to scroll vertically with table header fixed, and able to scroll vertically. But it is still a bit out of page.

5. REFLECTION AND DISCUSSION

5.1. PROFESSIONAL PURPOSE:

In our web development project, I was endowed with the responsibility of integrating PHP to reuse common elements and implement various functionalities such as listing EOIs, updating the status of an

EOI, and more. This role reemphasized the importance of clarity and purpose in my work. As a developer, I realized that the features we implement aren't just lines of code, but tools that can significantly affect the end user's experience. The purpose of our project was not just about creating a working website but ensuring that its backend functionality met the specific requirements. Every line of code, every design element, every functionality, had a reason and a purpose behind it. This project reminded me that in the professional world, we are not just tasked with doing things right but doing the right things. This project also made me appreciate the nuances of working in a real-world scenario where every decision has a ripple effect on other parts of the system.

5.2.TEAMWORK:

Teamwork, as always, came with its set of challenges and rewards. Starting the assignment three weeks before the deadline could have been a huge setback, but our team's commitment and drive turned it into a motivation. We worked faster, more efficiently, and with a sense of urgency that might not have been there otherwise. The camaraderie among group members was palpable. Weekend meetings at the state library, constant discussions, and troubleshooting sessions brought us closer, not just as teammates but as friends. Our shared commitment was evident in our determination to overcome challenges like the table scrolling issue in manage.php.

Being the group leader added another layer of responsibility. My role wasn't just about coding; it was about ensuring that every team member had a clear understanding of their tasks, facilitating resource sharing, and addressing challenges head-on. Coordinating the group, distributing tasks, and offering guidance were all crucial components of our project's success. This role taught me the value of delegation and trust. I learned that a leader's role is not just to oversee, but to empower each member to bring their best to the table. We faced challenges, notably our late start and issues with server access, which tested our resilience and adaptability. These challenges taught me the importance of proactive communication, timely check-ins, and contingency planning.

5.3. EFFECTIVE COMMUNICATION:

Effective communication was the backbone of our project. As the group leader, my doors were always open for questions, resource sharing, and problem-solving. Open lines of communication ensured that we were on the same page, even if we were working on different parts of the project. Challenges like the login.php issue and processEOI.php could have been major roadblocks, but our willingness to communicate, brainstorm, and find alternative solutions (like the sorry.php) ensured that we didn't get stuck.

5.4 WEB DEVELOMENT RELATED ISSUES

5.4.1 PRIVACY:

In our project, particularly in the login functionality, we were entrusted with user data. This experience made us deeply aware of the gravity of data privacy. We should ensured that data was encrypted and stored securely prevent leakage of applicants' personal information. In a world where data breaches are commonplace, I realized that as developers, we're not just coding; we're also gatekeepers of user trust. In future projects, I will ensure that data privacy is always a priority, adhering to best practices and standards to safeguard user information.

5.4.2 SECURITY:

Our challenges with the login functionality, especially implementing the disable login feature, highlighted the importance of security. While we opted for a workaround using 'sorry.php', it drove home the significance of secure coding practices. Web developers play a critical role in ensuring that vulnerabilities are minimized. I have resolved to continuously update my knowledge on web security, ensuring that every project I undertake is fortified against potential threats.

6. CONCLUSION

In conclusion, this report offers a detailed description of the SWIFT career website application submission page and manager user interface, highlighting its functionality and innovative elements. It has provided a comprehensive breakdown of each web page's primary content, highlighting the purpose and significance of each section. Furthermore, the report includes comprehensive research on the improving security of the website. This project was more than just a coding task; it was a deep dive into the intricacies of professional development, teamwork, and the ethical considerations of web development. The challenges we faced have enriched my understanding and appreciation of the web development process. It has inspired me to approach future projects with a renewed sense of purpose, commitment to team dynamics, and a deep respect for the privacy and security of users.

7. REFERENCE

Krombholz, K., Busse, K., Pfeffer, K., Smith, M. and Von Zezschwitz, E., 2019, May. " If HTTPS Were Secure, I Wouldn't Need 2FA"-End User and Administrator Mental Models of HTTPS. In *2019 IEEE Symposium on Security and Privacy (SP)* (pp. 246-263). IEEE.