Assignment Part 2 Server-Side Programming

M Anzor Yousuf
102849043
COS10026 Computing Technology Inquiry Project

Table of Contents

Introduction

In today's changing world of web development, creating a dynamic and user-friendly website is crucial for businesses to thrive in the digital age. The Cold Pies project showcases the journey of developing a comprehensive website that not only meets technical requirements but also enhances user experience and operational efficiency. This document outlines the step-by-step process undertaken to develop the Cold Pies website. The project involved setting up a development environment, designing a structured website, integrating a database for managing Expressions of Interest (EOI), and implementing user authentication for secure access control. The scope of the project included extending the functionality of the website by creating server-side PHP scripts to process job application data, creating simple MySQL tables for storing, updating, and retrieving information, and creating a web page for the Human Resources (HR) manager to manage job applications. The development was guided by best practices in web design and database management to deliver a seamless and functional web application. This report aims to provide a detailed account of the Cold Pies website development project, highlighting the methodologies, tools, and technologies used. It will cover the website's main functionalities, the security measures implemented, and the contributions made by the developer. Furthermore, it will also reflect on the challenges encountered and the solutions devised to overcome them, ensuring a successful project completion.

The Website

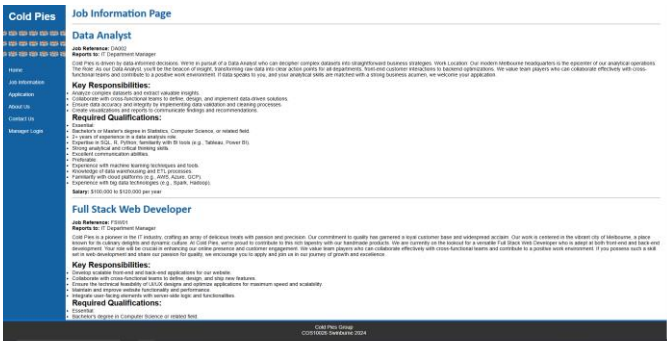
Website Introduction:

The Cold Pies website was developed to serve as a comprehensive platform for managing job applications and providing detailed information about available positions. The website is designed to cater to both potential applicants and HR managers, ensuring a seamless experience for submitting and managing job applications.

Main Functionalities of the Website:

1. Job Information Display:

- The website provides a detailed list of available job positions, including job titles, descriptions, key responsibilities, required qualifications, and salary ranges.
- Job information is dynamically retrieved from the database, ensuring that the content is always up to date.



Job Information Page

Data Analyst

Job Reference: DA002
Reports to: IT Department Manager

Cold Pies is driven by data-informed decisions. We're in pursuit of a Data Analyst who can decipher complex datasets into straightforward business strategies. Work Location: Our modern Melbourne headquarters is the epicenter of our analytical operations. The Role: As our Data Analyst, you'll be the beacon of insight, transforming raw data into clear action points for all departments, front-end customer interactions to backend optimizations. We value team players who can collaborate effectively with cross-functional teams and contribute to a positive work environment. If data speaks to you, and your analytical skills are matched with a strong business acumen, we welcome your application.

Key Responsibilities:
- Analyze complex datasets and extract valuable insights.
- Collaborate with cross-functional teams to define, design, and implement data-driven solutions.
- Ensure data accuracy and integrity by implementing data validation and cleaning processes.
- Create visualizations and reports to communicate findings and recommendations.

Required Qualifications:
- Essential:
- Bachelor's or Master's degree in Statistics, Computer Science, or related field.
- 2+ years of experience in a data analysis role.
- Expertise in SQL, R, Python; familiarity with BI tools (e.g., Tableau, Power BI).
- Strong analytical and critical thinking skills.
- Excellent communication abilities.
- Preferable:
- Experience with machine learning techniques and tools.
- Knowledge of data warehousing and ETL processes.
- Familiarity with cloud platforms (e.g., AWS, Azure, GCP).
- Experience with big data technologies (e.g., Spark, Hadoop).

Salary: $100,000 to $120,000 per year

Full Stack Web Developer

Job Reference: FSW01
Reports to: IT Department Manager

Cold Pies is a pioneer in the IT industry, crafting an array of delicious treats with passion and precision. Our commitment to quality has garnered a loyal customer base and widespread acclaim. Our work is centered in the vibrant city of Melbourne, a place known for its culinary delights and dynamic culture. At Cold Pies, we're proud to contribute to this rich tapestry with our handmade products. We are currently on the lookout for a versatile Full Stack Web Developer who is adept at both front-end and back-end development. Your role will be crucial in enhancing our online presence and customer engagement. We value team players who can collaborate effectively with cross-functional teams and contribute to a positive work environment. If you possess such a skill set in web development and share our passion for quality, we encourage you to apply and join us in our journey of growth and excellence.

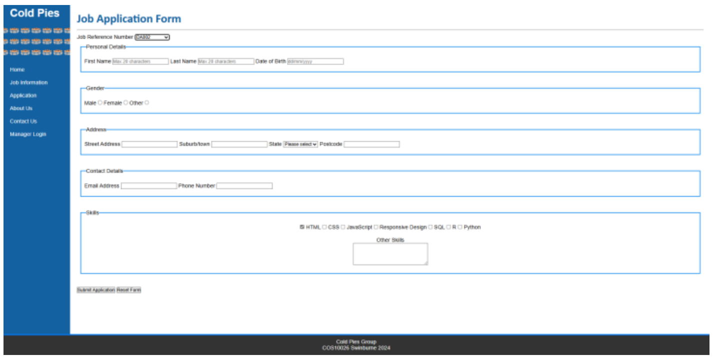
Key Responsibilities:
- Develop scalable front-end and back-end applications for our website.
- Collaborate with cross-functional teams to define, design, and ship new features.
- Ensure the technical feasibility of UI/UX designs and optimize applications for maximum speed and scalability.
- Maintain and improve website functionality and performance.
- Integrate user-facing elements with server-side logic and functionalities.

Required Qualifications:
- Essential:
- Bachelor's degree in Computer Science or related field.

Cold Pies Group
COS10026 Swinburne 2024

2. Application Form:

- The application form allows users to submit their Expressions of Interest (EOI) for specific job positions.
- The form includes fields for personal details, contact information, and skills, along with options to select job reference numbers from a dropdown list.
- Server-side validation ensures that all required fields are properly filled out before submission.



Cold Pies

Home
Job Information
Application
About Us
Contact Us
Manager Login

Job Application Form

Job Reference Number DA002

Personal Details
First Name Max 20 characters Last Name Max 20 characters Date of Birth dd/mm/yyyy

Gender
Male ○ Female ○ Other ○

Address
Street Address Suburb/town State Please select Postcode

Contact Details
Email Address Phone Number

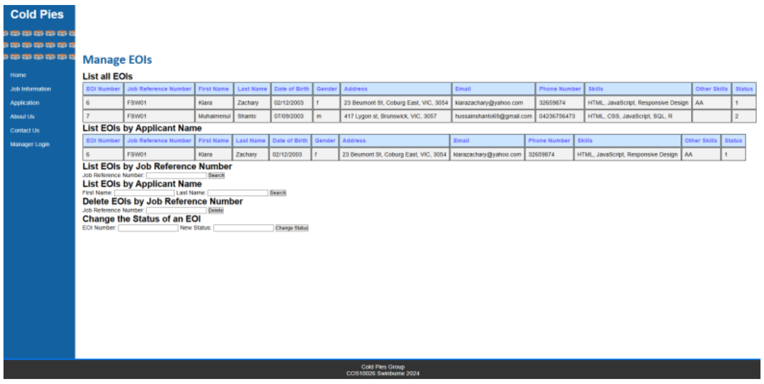
Skills
☑ HTML ☐ CSS ☐ JavaScript ☐ Responsive Design ☐ SQL ☐ R ☐ Python
Other Skills

Submit Application Reset Form

Cold Pies Group
COS10026 Swinburne 2024

3. Manager's Page:

- A dedicated page for HR managers to manage submitted EOIs.
- Managers can view all submitted EOIs, filter them by job reference numbers or applicant names, delete specific EOIs, and update the status of EOIs.
- This functionality is secured by user authentication, ensuring that only authorized personnel can access and manage the EOIs.



Cold Pies

Home
Job Information
Application
About Us
Contact Us
Manager Login

Manage EOIs

List all EOIs

EOI Number	Job Reference Number	First Name	Last Name	Date of Birth	Gender	Address	Email	Phone Number	Skills	Other Skills	Status
6	FSW01	Kiara	Zachary	02/12/2003	f	23 Beumont St, Coburg East, VIC, 3054	kiarazachary@yahoo.com	32659874	HTML, JavaScript, Responsive Design	AA	1
7	FSW01	Muhaimenul	Shanto	07/09/2003	m	417 Lygon st, Brunswick, VIC, 3057	hussainshanto69@gmail.com	04236756473	HTML, CSS, JavaScript, SQL, R		2

List EOIs by Applicant Name

EOI Number	Job Reference Number	First Name	Last Name	Date of Birth	Gender	Address	Email	Phone Number	Skills	Other Skills	Status
6	FSW01	Kiara	Zachary	02/12/2003	f	23 Beumont St, Coburg East, VIC, 3054	kiarazachary@yahoo.com	32659874	HTML, JavaScript, Responsive Design	AA	1

List EOIs by Job Reference Number
Job Reference Number: Search

List EOIs by Applicant Name
First Name: Last Name: Search

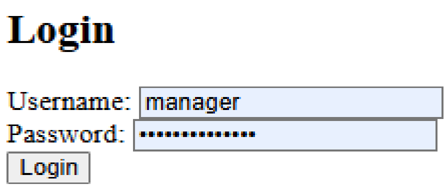
Delete EOIs by Job Reference Number
Job Reference Number: Delete

Change the Status of an EOI
EOI Number: New Status: Change Status

Cold Pies Group
COS10026 Swinburne 2024

4. User Authentication:

- Implemented login functionality for HR managers to securely access the management page.
- Authentication ensures that only authorized users can perform sensitive operations such as deleting or updating EOIs.



Login

Username: manager
Password: •••••••••••••••
Login

Technical Details on the Website Development:

1. Development Environment Setup:

- Apache web server and PHP were installed on a local machine to create a development environment.
- MySQL was used as the database management system, with a new database and user account created for the project.

2. Website Structure:

- The website's folder structure was organized with separate directories for HTML, CSS, PHP, and images.
- Common elements like headers, menus, and footers were included using PHP includes to ensure consistency and maintainability across all pages.

3. Database Integration:

- The "eoi" table was created in the MySQL database to store job application data, including fields for EOInumber, JobReferenceNumber, FirstName, LastName, DateOfBirth, Gender, Address, Email, PhoneNumber, Skills, OtherSkills, and Status.
- The EOInumber column was set as an auto-incrementing primary key to uniquely identify each record.

4. Server-Side Scripting:

- PHP scripts were developed to handle form submissions, validate data, and interact with the MySQL database.
- The processEOI.php script validates form data and inserts new EOIs into the database, setting the status to "New" for each entry.
- If the "eoi" table does not exist, it is created programmatically before inserting records.

5. Security Measures:

- Input validation and sanitization were implemented to prevent SQL injection and other common security vulnerabilities.
- User authentication ensures that only authorized HR managers can access and manipulate job application data.

Key and Innovative Features of the Website:

1. Dynamic Content Generation:

- Job information and EOIs are dynamically fetched from the database, allowing for real-time updates and reducing the need for manual content management.

2. Secure User Authentication:

- Implementation of a secure login system for HR managers ensures that sensitive operations are protected and only accessible by authorized users.

3. User-Friendly Interface:

- The website's design focuses on providing a clear and intuitive interface for both job applicants and HR managers.
- The use of PHP includes, and CSS ensures a consistent and visually appealing layout across all pages.

4. Enhanced Functionality for HR Managers:

- The manager's page provides comprehensive tools for managing EOIs, including filtering, deleting, and updating records, streamlining the HR management process.
- By combining these functionalities and technical details, the Cold Pies website offers a robust and efficient platform for managing job applications, ensuring both ease of use and security.

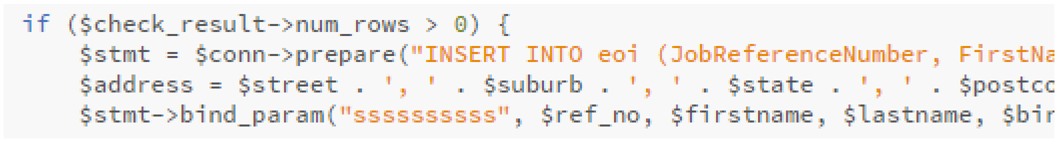
Security of the Website

To enhance the security of the Cold Pies website, various measures were researched and implemented.

Implementation of Security Tips in Code:

- *Prepared Statements and Parameterized Queries:*

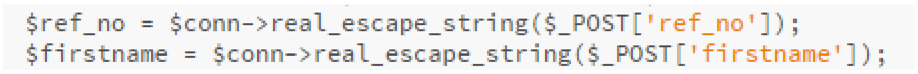
Throughout the website, prepared statements with parameterized queries have been used to prevent SQL injection attacks. This is particularly important in scripts that interact with the database, such as 'processEOI.php' and 'manage.php'.



```
if ($check_result->num_rows > 0) {
    $stmt = $conn->prepare("INSERT INTO eoi (JobReferenceNumber, FirstNa
    $address = $street . ', ' . $suburb . ', ' . $state . ', ' . $postcc
    $stmt->bind_param("ssssssssss", $ref_no, $firstname, $lastname, $bir
```

- *Form Validation and Sanitization:*

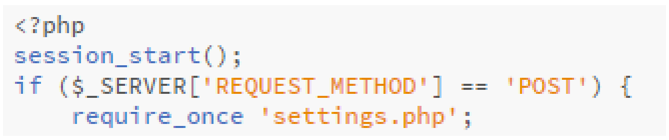
Both client-side and server-side validations are implemented to ensure that the input data is in the correct format and sanitized before being processed. For example, in 'processEOI.php', user inputs are sanitized using 'real_escape_string()' to prevent SQL injection.



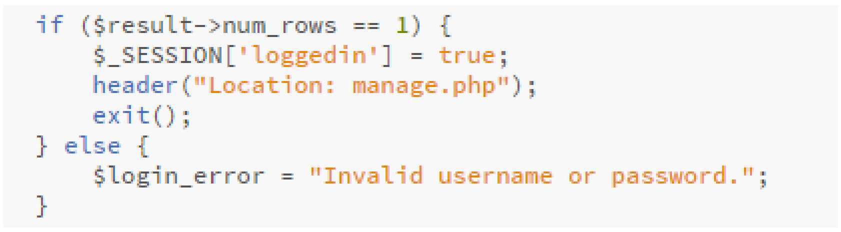
```
$ref_no = $conn->real_escape_string($_POST['ref_no']);
$firstname = $conn->real_escape_string($_POST['firstname']);
```

- *Session Management:*

Secure session handling is implemented to manage user authentication. Sessions are started at the beginning of PHP scripts requiring authentication and checked to ensure the user is logged in.



```
<?php
session_start();
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    require_once 'settings.php';
```



```php
if ($result->num_rows == 1) {
    $_SESSION['loggedin'] = true;
    header("Location: manage.php");
    exit();
} else {
    $login_error = "Invalid username or password.";
}
```

Improvements that can be implemented:

- *Password Hashing:*

  In ‘phpenhancements.php’, currently, passwords are stored in plain text, which is a significant security risk. By hashing passwords using a secure hashing algorithm such as bcrypt before storing them in the database, you ensure that even if the database is compromised, attackers cannot retrieve the original passwords. Password hashing converts passwords into a fixed-length hash that is not reversible. Even if someone gains access to the database, they cannot easily convert the hash back to the original password. This is especially critical because users often reuse passwords across multiple sites, so protecting their password on your site helps protect their accounts on other sites too.

- *HTTPS for Secure Communication:*

  While the code itself does not directly implement HTTPS, configuring the web server to use HTTPS ensures all data transferred between the user and the server is encrypted. This prevents eavesdropping and man-in-the-middle attacks. HTTPS encrypts the data sent between the user's browser and your web server, protecting sensitive information like login credentials, personal data, and other confidential data from being intercepted by malicious actors.

- *Cross-Site Scripting (XSS) Protection:*

  To prevent XSS attacks, you should sanitize and escape user inputs before displaying them on the web page. In ‘processEOI.php’ and ‘manage.php’, this can be done using PHP functions like ‘htmlspecialchars()’. XSS attacks inject malicious scripts into web pages viewed by other users. By sanitizing and escaping user inputs, you prevent attackers from executing malicious scripts that can steal cookies, session tokens, or other sensitive information.

- *Cross-Site Request Forgery (CSRF) Protection:*

  To protect against CSRF attacks, include a hidden CSRF token in forms and verify this token upon form submission. This can be added to the ‘apply.php’ form and verified in ‘processEOI.php’. CSRF attacks trick users into performing actions they did not intend by exploiting their authenticated session with a trusted site. By using a CSRF token, you ensure that form submissions are genuine and intended by the user.

- *Database User with Limited Privileges:*

  Ensure that the database user used in ‘settings.php’ has only the necessary privileges for the application. For example, a read-only user for fetching data and a separate user with insert/update/delete permissions for modifying data. Limiting database user privileges reduces the risk of an attacker gaining full control of the database if they manage to exploit an SQL injection vulnerability or other weaknesses.

My Contribution

In this project, my primary contributions included:

1. Setting Up the Development Environment:

a. Installed Apache server, PHP, and MySQL on my local machine to ensure a suitable environment for web development.
b. Configured the database and user permissions required for the project.

2. Creating the Website Structure:

a. Designed a well-organized folder structure to separate HTML, CSS, PHP, and images.
b. Developed the main HTML pages (index.php, jobs.php, apply.php, about.php) and utilized PHP includes for headers, menus, and footers to maintain consistency and modularity across the website.

3. Developing the EOI Table:

a. Connected to the MySQL database and created the eoi table with necessary columns such as EOInumber, Job Reference number, First name, Last name, Address, Email, Phone number, Skills, Other skills, and Status.
b. Ensured the EOInumber column was set as an auto-incrementing primary key.

4. Creating the Application Form:

a. Modified the application form from a previous assignment to send form data to processEOI.php.
b. Implemented server-side validation for form data and inserted valid entries into the eoi table.
c. Provided feedback to users with confirmation messages displaying the auto generated EOInumber.

5. Developing the HR Manager Page:

a. Created manage.php to allow HR managers to view, update, and delete EOIs.
b. Implemented functionalities to list EOIs by various criteria and update EOI statuses.

6. Enhancements:

a. Stored job descriptions in a separate database table and dynamically generated HTML using PHP.
b. Normalized the database structure by creating primary-foreign key relationships between tables.
c. Added user authentication and access control for the manager's page.
d. Implemented a logout functionality for the manager.

Reflection and Discussion

Professional Purpose, Teamwork, and Effective Communication

Working on this project provided me with significant insights into the importance of professional purpose, teamwork, and effective communication. Despite being part of a group initially, I was excluded just before this assignment and had to complete the project independently. This situation posed a considerable challenge but also a valuable learning experience.

Challenges Faced:

Team Dynamics: Being excluded from the group meant I had to take on all responsibilities, which was daunting. It highlighted the importance of inclusive team dynamics and the need for effective conflict resolution strategies.

Time Management: Juggling multiple roles required time management. I had to plan and execute each phase systematically to ensure timely completion.

Problem-Solving: Encountering issues such as database connection errors and validation problems required persistent problem-solving and debugging skills.

Key Learnings:

Resilience and Adaptability: Navigating through the project alone reinforced my resilience and adaptability. It showed that with determination and a clear plan, challenges could be overcome.

Technical Skills: The hands-on experience enhanced my technical skills in PHP, MySQL, and web development. I learned to create a seamless, user-friendly interface and implement robust backend functionalities.

Web Development Related Issues: Security

Data Validation and Sanitization: Ensuring all user inputs were validated and sanitized to prevent SQL injection and cross-site scripting (XSS) attacks.

Password Hashing: Implementing secure password hashing techniques for user authentication.

HTTPS: Using HTTPS to encrypt data transmitted between the server and clients.

Access Control: Restricting access to sensitive pages such as manage.php to authorized users only.

Regular Updates: Keeping the server and software up to date with the latest security patches.

These security measures not only safeguarded the application but also built user trust by protecting their data.

Conclusion

In conclusion, this project was a comprehensive exercise in web development, encompassing both frontend and backend development, database management, and security implementation. The challenges faced and overcome during this project significantly enhanced my technical and soft skills, preparing me for the future in the tech industry. The project successfully met the requirements, providing a functional job application and management system with enhanced security features.

Reference

1. Arias, D. (2018). *Hashing Passwords: One-Way Road to Security*. [online] Auth0 - Blog. Available at: https://auth0.com/blog/hashing-passwords-one-way-road-to-security/.
2. Cloudflare (n.d.). What is HTTPS?
   | Cloudflare UK. *Cloudflare*. [online] Available at: https://www.cloudflare.com/en-gb/learning/ssl/what-is-https/.
3. James, N. (2023). *Security Audit : an Expert Guide*. [online] Astra. Available at: https://www.getastra.com/blog/security-audit/security-audits/.
4. Kime, C. (2023). *7 database security best practices | eSecurity planet*. [online] ESecurityPlanet. Available at: https://www.esecurityplanet.com/networks/database-security-best-practices/.
5. Maury, J. (2023). *How to Prevent Web Attacks Using Input Sanitization*. [online] eSecurityPlanet. Available at: https://www.esecurityplanet.com/endpoint/prevent-web-attacks-using-input-sanitization/.
6. OWASP (2012). *Cross-Site Request Forgery Prevention · OWASP Cheat Sheet Series*. [online] Owasp.org. Available at: https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html.
7. OWASP (2020). *Improper Error Handling | OWASP*. [online] Owasp.org. Available at: https://owasp.org/www-community/Improper_Error_Handling.
8. OWASP (2021). *SQL Injection Prevention · OWASP Cheat Sheet Series*. [online] Owasp.org. Available at: https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html.
9. PortSwigger (2023). *What is cross-site scripting (XSS) and how to prevent it?* [online] Portswigger.net. Available at: https://portswigger.net/web-security/cross-site-scripting.
10. Psinas, M. (2012). *Role Based Access Control in PHP*. [online] SitePoint. Available at: https://www.sitepoint.com/role-based-access-control-in-php/.