 **Assignment Cover Sheet**

|  |  |  |
| --- | --- | --- |
| **Student Information (For group assignment, please state names of all members)** | | **Grade/Marks** |
| **Name: ISAAC YUEN XI SHENG** | **ID: B2400818** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Module/Subject Information** | | **Office Acknowledgement** |
| **Module/Subject Code** | DIP107/1009 |  |
| **Module/Subject Name** | FUNDAMENTALS OF WEB DEVELOPMENT |  |
| **Lecturer/Tutor/Facilitator** | MISS.NALINE SHANMUGAM |  |
| **Due Date** | 21 APRIL 2025 |  |
| **Assignment Title/Topic** | BACKEND DEVELOPMENT |  |
| **Intake (where applicable)** |  |  |
| **Word Count** |  | **Date/Time** |

**Declaration**

* I/We have read and understood the Programme Handbook that explains on **plagiarism**, and I/we testify that, unless otherwise acknowledged, the work submitted herein is entirely my/our own.
* I/We declare that no part of this assignment has been written for me/us by any other person(s) except where such collaboration has been authorized by the lecturer concerned.
* I/We authorize the University to test any work submitted by me/us, using text comparison software, for instances of plagiarism. I/We understand this will involve the University or its contractors copying my/our work and storing it on a database to be used in future to test work submitted by others.

Note: 1) The attachment of this statement on any electronically submitted assignments will be deemed to have the same authority as a signed statement.

2) The Group Leader signs the declaration on behalf of all members.

|  |  |
| --- | --- |
| Signature: | Date:20 April , 2025 |
| E-mail: |  |

**INTRODUCTION**

The objective of this project was to add backend functionality to our Assignment 1 project. I chose to do the contact form feature because it is a very common part of most websites. The objective was to allow users to send messages into the database. And for the website to save those messages into the database. This is including creating and connecting a form, saving data, and creating an admin panel to view, edit, and delete those messages.

I chose the contact form because it is a function that enables users to engage directly with the site. If there is no backend, the form would be useless. By linking it to the database, it becomes a helpful tool for gathering messages, feedback, and even support requests from users. This also helped me learn how dynamic websites function in the background.

**TECHNOLOGY USED**

For the front end, I use basic HTML to create the contact form. For the backend, I used PHP to process the form data and interact with the database. The database was created using MySQL, and I managed it using phpMyAdmin. The project was run by XAMPP.

XAMPP helped me run a web server and database server locally on my laptop. Apache helped me run the PHP code and test it, and MySQL managed all the form submissions. phpMyAdmin helped me to view the table and information with ease without the need to enter SQL manually. All these components helped me comprehend better how full-stack development works.

**DATABASE DESIGN**

The table that was created is called contact\_messages. This table includes

* Id
* Name
* Email
* Subject
* Message

Each column holds a component of the form submission. The ID is auto-generated and set as the primary key. The remaining columns are completed by users on the contact form. This table was created through phpMyAdmin and resides under the database assignment2. This design can be applied in real-world usage in gathering complaints, feedback, or business queries. This layout is simple but is extremely effective in storing messages.

**BACKEND DEVELOPMENT**

**Step 1: CREATING THE FORM**

There is a file called contact\_form.html with input fields for name, email, subject, and message. The form use the POST method and submitted the data to contact.php .

This is the main entry for the users to post their messages. All fields are made compulsory so that no messages is posted with incomplete details.

**Step 2: HANDLING FORM DATA WITH PHP**

In contact.php, MySQL database is connected using PHP and inserted the submit form data into the contact\_messages table.

Basic validation was included to make sure all the inputs were filled. On submission, the PHP file saves the data and shows a success message. This file is the main interface between the form and database.

**Step 3: VIEWING MESSAGES**

I have created a page view\_messages.php to show all the messages in a table. Each row of messages includes an Edit and Delete option.

This page is useful to admin users because it shows all the form submissions at one place. The messages are shown in reverse chronological order (newest first).

**Step 4: DELETING MESSAGES**

By using delete\_message.php, users are allowed to remove messages from the database using message ID.

The delete button asks for confirmation before deleting the message. When deleted, the user is redirected back to the list of messages.

**Step 5: EDITING MESSAGES**

With edit\_messages.php, users are allowed to edit and update messages and save the changes to the database.

This feature allows the chosen message to be opened in a form with the existing information preentered. When the updates are complete, the new data gets stored in the same database row.

**CHALLENGES FACED**

* I had some trouble at first because the form did not connect to the database properly because the file had a wrong name. It turns out I had saved it as a .txt by accident
* I had trouble seeing the submitted data until I learned how to use phpMyAdmin properly, thanks to some google searches and using AI tools to help me out.
* I didn’t see form results at first because I didn’t have a success message.

**TESTING AND DEBUGGING**

I have tested the form by filling it out multiple times and checking if data was visible in phpMyAdmin. I have also tested the edit and delete function by clicking on the links and checking if any changes happen. Each link and page was tested on a localhost in my browser, and I made sure that Apache and MySQL was running constantly on XAMPP.

I also tested each submission form after to make sure that the message was saved correctly. I also tested delete and edit. I tested all the pages a minimum of three times and worked perfectly once the first set of bugs were fixed.

**CONCLUSION**

This project help me understand how to connect the frontend and the backend using PHP and MySQL. I also learned how to create, read, update, and delete data using a form. I also learned how to do debugging and testing on my local server.

|  |
| --- |
| **Feedback/Comments\*** |
| **Main Strengths** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **Main Weaknesses** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **Suggestions for improvement** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |  |
| --- | --- |
|  | **Student acknowledge feedback/comments** |
|  |
| Grader’s signature | Student’s signature: |
| Date: | Date: |

Note:

1. A soft and hard copy of the assignment shall be submitted.
2. The signed copy of the assignment cover sheet shall be retained by the marker.
3. If the Turnitin report is required, students have to submit it with the assignment. However, departments may allow students up to **THREE** (3) working days after submission of the assignment to submit the Turnitin report. The assignment shall only be marked upon the submission of the Turnitin report.

\*Use additional sheets if required.