

Course: IT202-008-S2025

Assignment: IT202 Module 4 SQL Challenge

Student: Gori H. (gs658)

Status: Submitted | Worksheet Progress: 106%

Potential Grade: 10.00/10.00 (100.00%)

Received Grade: 0.00/10.00 (0.00%)

Grading Link: <https://learn.ethereallab.app/assignment/v3/IT202-008-S2025/it202-module-4-sql-challenge/grading/gs658>

Instructions

1. Ensure you read all instructions and objectives before starting.

2. Create a new branch from dev called M4-Homework

1. git checkout dev (ensure proper starting branch)
2. git pull origin dev (ensure history is up to date)
3. git checkout -b M4-Homework (create and switch to branch)

3. Copy the template code from here: [GitHub Repository - M4 Homework](#)

- Put all into an M4 folder inside your public_html
- Expected files

- sql/000_create_table.todos.sql
- sql/init_db.php
- todos/create.php
- todos/pending.php
- todos/completed.php
- nav.php

- Immediately record to history
 - git add public_html
 - git commit -m "adding M4 HW baseline files"
 - git push origin M4-Homework
 - Create a Pull Request from M4-Homework to dev and keep it open

4. Fill out the below worksheet

- Each Problem requires the following as you work

- Ensure there's a comment with your UCID, date, and brief summary of how the problem was solved
- Update ucid in nav.php where noted
- Code solution (add/commit periodically as needed) (refer to comments in the create, pending, completed files)

5. Once finished, click "Submit and Export"

6. Locally add the generated PDF to a folder of your choosing inside your repository folder and move it to Github

1. git add .
2. git commit -m "adding PDF"
3. git push origin M4-Homework
4. On Github merge the pull request from M4-Homework to dev
5. On Github create a pull request from dev to prod and immediately merge. (This will trigger the prod deploy to make the heroku prod links work)

7. Upload the same PDF to Canvas

8. Sync Local

1. git checkout dev
2. git pull origin dev

100%

Section #1: (1 pt.) Prep

Details:

Once the files are copied in, run your local PHP server and visit /M4/sql/init_db.php in the browser to trigger the table generation.

100%

Task #1 (0.50 pts.) - Show the output of the init_db.php tool

Image Prompt

Weight: 50%

Objective: *Show the output of the init_db.php tool*

Details:

The output should note either successfully executed or blocked (this is ok)



Database Helper Tool

Warning: /app/public_html/M4/sql/000_create_table_toolbox.sql
Blocked from running, table found in 'show tables' results. [This is ok, it reduces redundant EDD calls]

Init complete, used approximately 1 db calls.

showing init_db.php success/block



Saved: 5/2/2025 10:03:00 PM

100%

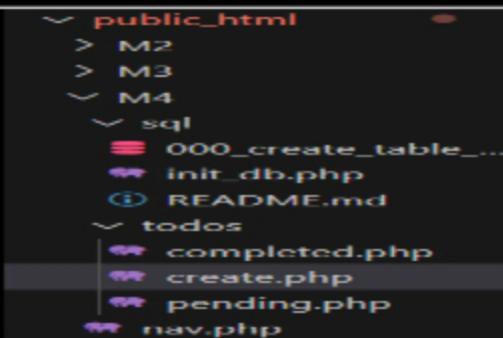
Task #2 (0.50 pts.) - Show the project view of your editor to show a

Image Prompt

Image Prompt

Weight: 50%

Objective: Show the project view of your editor to show all files are present



```
public_html
  > M2
  > M3
  < M4
    < sql
      000_create_table_...
      init_db.php
      README.md
    < todos
      completed.php
      create.php
      pending.php
      nav.php
index.php
```

showing required files in vs code



Saved: 5/2/2025 10:04:34 PM

100%

Section #2: (3 pts.) Create Page

100%

Task #1 (0.75 pts.) - HTML Form Challenge

Combo Task:

Weight: 25%

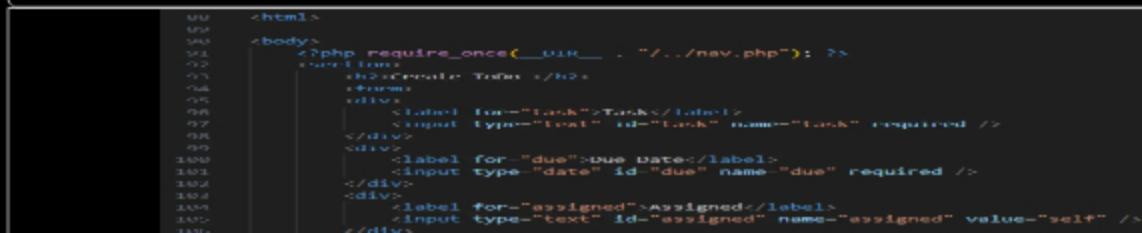
Objective: HTML Form Challenge

Image Prompt

Weight: 50%

Details:

Show the code snippet related to the completed HTML form



```
<html>
  <body>
    <?php require_once __DIR__ . "/../nav.php"; ?>
    <h1>Todos</h1>
    <ol>
      <li>Todos</li>
      <li>Completed</li>
      <li>Pending</li>
    </ol>
    <form>
      <label for="due">Due Date</label>
      <input type="date" id="due" name="due" required />
    </form>
    <form>
      <label for="assigned">Assigned</label>
      <input type="text" id="assigned" name="assigned" value="self" />
    </form>
  </body>
</html>
```

showing the form fields

↓



Saved: 5/2/2025 10:11:13 PM

≡, Text Prompt

Weight: 50%

Details:

Items

- Note the field names and why the specific types were chosen
- Also note how you made "self" the default for the assigned input

Your Response:

The form fields are set with different types to match the SQL schema. Task uses text for longer descriptions, due uses date for valid date inputs, and assigned uses text to store the task name. The assigned field defaults to "self" using the value = "self" attribute to make sure tasks are auto assigned to the user submitting the form.



Saved: 5/2/2025 10:11:13 PM



Task #2 (0.75 pts.) - Form Submission PHP Validations

Combo Task:

Weight: 25%

Objective: *Form Submission PHP Validations*

≡, Image Prompt

Weight: 50%

Details:

- Validate the incoming data for correct format based on the SQL table definition.
- When not valid, provide a user-friendly message of what specifically was wrong and set `$is_valid` to false

- Assigned should check for "self" if a valid format/value isn't provided
- Include screenshot(s) from Heroku Dev demonstrating each validation (ensure URL is visible)

```

if (empty($GET['task'])) {
    // data variables, don't want
    // task = & GET['task'] // this would be a valid MySQL value format
    $task = & GET['task'] // this would be a valid MySQL value format
    $assigned = & GET['assigned'] // this would be a valid MySQL value format
    $isValid = true;
    // check if the incoming data has a valid format based on the SQL rule
    // if not valid, possible a user friendly message will show up differently
    // assigned should check for "self" if a valid format/value isn't provided
    // start validations
    // mm/dd/yyyy This validation ensures the task is not empty and that
    // due date is in the correct format. It also ensures the assigned field
    // doesn't say "self" when left blank
    // 
    if (empty($task)) {
        echo "Task is required";
        $isValid = false;
    }

    $date = DateTime::createFromFormat('Y-m-d', $due);
    if ($date == null) {
        echo "Due date must be in YYYY-MM-DD format";
        $isValid = false;
    }

    if (empty($assigned) || $assigned != "self") {
        $assigned = "self";
    }
    // End validation
}

```

showing validation logic

gsx58 | Friday, May 2, 2025 at 10:16 PM

Create ToDo

Task

Due Date
mm//dd/yyyy

Assigned
self

Task is required.

Create ToDo Pending ToDos Completed ToDos

gsx58 | Friday, May 2, 2025 at 10:16 PM

Create ToDo

Task

Due Date
mm//dd/yyyy

Assigned
self

showing incorrect date format output

gsx58 | Friday, May 2, 2025 at 10:23 PM

Create ToDo

Task

Due Date
mm//dd/yyyy

Assigned
self

Task is required.

Create ToDo Pending ToDos Completed ToDos

gsx58 | Friday, May 2, 2025 at 10:23 PM

Create ToDo

Task

Due Date
mm//dd/yyyy

Assigned
self

showing empty task output

Saved: 5/2/2025 10:29:07 PM

☒ Text Prompt

Weight: 50%

Details:

Briefly explain your validation logic.

100%

Task #3 (0.75 pts.) - Show the code snippet related to the INSERT c

Combo Task:

Weight: 25%

Objective: *Show the code snippet related to the INSERT query*

☞ Image Prompt

Weight: 50%

Details:

Design a query to insert the incoming data to the proper columns. Ensure valid and proper PDO named placeholders are used. <https://phpdelusions.net/pdo>

```
21  // $name = $_POST['name'];
22  // $email = $_POST['email'];
23  // $password = password_hash($_POST['password'], PASSWORD_DEFAULT);
24  // $image = $_FILES['image'];
25  // $category_id = $_POST['category_id'];
26  // $status = $_POST['status'];
27  // $description = $_POST['description'];
28  // $tags = $_POST['tags'];
29  // $tags = str_replace(' ', ',', $tags);
30  // $tags = explode(',', $tags);
31  // $tags = array_map('trim', $tags);
32  // $tags = array_map('mb_convert_encoding', $tags, 'UTF-8');
33  // $tags = array_map('mb_ereg_replace', $tags, '/[^a-zA-Z0-9\s]/');
34  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
35  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
36  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
37  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
38  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
39  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
40  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
41  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
42  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
43  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
44  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
45  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
46  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
47  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
48  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
49  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
50  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
51  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
52  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
53  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
54  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
55  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
56  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
57  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
58  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
59  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
60  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
61  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
62  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
63  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
64  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
65  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
66  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
67  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
68  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
69  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
70  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
71  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
72  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
73  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
74  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
75  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
76  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
77  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
78  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
79  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
80  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
81  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
82  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
83  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
84  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
85  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
86  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
87  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
88  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
89  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
90  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
91  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
92  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
93  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
94  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
95  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
96  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
97  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
98  // $tags = array_map('mb_ereg_replace', $tags, '/\s+/');
99  // $tags = array_map('mb_ereg_replace', $tags, '/^\s+|\s+$/');
```

showing my insert query



Saved: 5/2/2025 10:33:43 PM

☞ Text Prompt

Weight: 50%

Details:

Briefly describe **how** the query works in plain words.

Your Response:

The query uses named placeholders to insert user input into the corresponding Todo table columns. This not only inserts the data correctly but also protects against SQL injection.

Saved: 5/2/2025 10:33:43 PM

Task #4 (+ 0.94 pts.) - Extra Credit: Handle Unique Constraint Errors

Combo Task:

Weight: 31.25%

Objective: Extra Credit: Handle Unique Constraint Errors

Image Prompt

Details:

Show the code related to handling the unique constraint errors and the user-friendly message

```
        } catch (PDOException $e) {
            // extra credit
            // check if the exception was related to a unique constraint
            // provide an appropriate user friendly message for this scenario
            // otherwise show the default message below
            /* GS658 - 5/2/25 - The check will identify if the PDO exception
            was caused by a unique constraint violation, and will display a
            user-friendly message. */
            if (strpos($e->getMessage(), 'UNIQUE') === false) {
                echo "This task already exists.";
            } else {
                echo "There was an error inserting the record. Check the logs.
                error_log("Insert Error: " . var_export($e, true));
            }
        }
    }
```

showing extra credit logic



Saved: 5/2/2025 10:36:16 PM

Text Prompt

Details:

Briefly explain **how** your code solves the challenge.

Your Response:

My code solves the challenge because the check will identify if the PDO exception was caused by a unique constraint violation, and will display a user-friendly message if so.



Saved: 5/2/2025 10:36:16 PM

Task #5 (0.75 pts.) - Create Page URLs

Url Prompt

Weight: 25%

Objective: *Create Page URLs*

Details:

- Direct link to the file in the homework related branch from Github (should end in `.php`)
- Direct link to the file on Heroku Prod (Just grab the base prod url and manually enter the path to the file)

URL #1

<https://gs658-it202-008->

prod-96409305ed03.herokuapp.com/todos/create.php



URL

<https://gs658-it202-008-prod-96409301>



URL #2

<https://github.com/goribantu/gs658->

[IT202-008-M4-](https://github.com/goribantu/gs658-IT202-008-M4-)



URL

<https://github.com/goribantu/gs658-IT202-008-M4->



100%

Section #3: (3 pts.) Pending Page

100%

Task #1 (1 pt.) - Fetching Todos

Combo Task:

Weight: 33.33%

Objective: *Fetching Todos*

Image Prompt

Weight: 50%

Details:

- Refer to the HTML table below and build a query that'll select the columns in the same order as the table from the Todo table
- Cross-reference the HTML table columns with what'd most plausibly match the SQL table aside from the notes below
- For the Status part, you'll need to calculate the "days_offset" from the due date, ensure the

virtual column matches "days_offset"

- For Actions, this isn't part of the query and there's nothing special to select for it
 - Filter the results where the todo item is NOT completed and order the results by those due the soonest
 - No limit is required
 - Include screenshot(s) from Heroku Dev demonstrating 5 different records (ensure URL is visible)

```

/* Refer to the HTML table below and build a query that'll select the columns
cross reference the HTML table columns with what'd most plausibly match the SQL
From the Status field, you'll need to calculate the "days offset" from the due/d
Due Actions. This isn't part of the query and there's nothing specific to select
filter the results where the todo item is NOT completed and order the results
no limit is required.
*/
/* gwtcm = 5/2/20 - this query will filter tasks that are not completed using
completed IN (0,1) and DATEOFFSET(due, CURDATE()) to calculate the days offset
ORDER BY due ASC will order the tasks by the nearest due date.
*/
Query = "SELECT Id, task, due, assigned, DATEOFFSET(due, CURDATE()) AS
days_offset FROM MM_todos WHERE completed IS NULL ORDER BY due ASC";
results = [];
try {
    Stmt = Sql.prepareStatement(Query);
    Stmt.setInt(resultset);
    if (Stmt != null) {
        results = Stmt.executeQuery();
    }
} catch (SQLException Re) {
    Re.printStackTrace();
    error_log("Please refresh pending todos, check the logs. (" + Re.getMessage() + ")");
    error_log("Select Error: " + Re.getLocalizedMessage()); // Shows in the terminal
}
}

```

showing fetch code

Pending ToDos						
ID	Task	Due Date	Status	Assigned	Actions	
6	clean room	2025-05-02	self	Overdue by 1 day(s)	Complete	
7	work on IT work	2025-05-02	self	Overdue by 1 day(s)	Complete	
4	dinner with mary	2025-05-05	self	Due in 2 day(s)	Complete	
5	go to the gym	2025-05-07	self	Due in 4 day(s)	Complete	
3	study for physics common	2025-05-10	self	Due in 7 day(s)	Complete	

showing correct output



Saved: 5/2/2025 10:45:04 PM

≡, Text Prompt

Weight: 50%

Details:

Briefly describe how the query works in plain words.

Your Response:

This query will filter tasks that are not completed using completed IS NULL and DATEDIFF(due, CURDATE()) to calculate the days_offset, and ORDER BY due ASC will order the tasks by the nearest due date.



Saved: 5/2/2025 10:45:04 PM



Task #2 (1 pt.) - Completing ToDos

Combo Task:

Weight: 33.33%

Objective: *Completing Todos*

≡, Image Prompt

Weight: 50%

Details:

- Create a query that'll update the respective ToDo marking it complete and setting the date for the completed date field as today
 - Ensure the "id" is utilized using proper PDO named placeholders so that only the one item is updated
 - Add an extra clause to update only if the complete field of the record is not set

showing how todos are completed properly



Saved: 5/2/2025 10:47:39 PM

Text Prompt

Weight: 50%

Details:

Briefly describe how the query works in plain words.

Your Response:

The query marks tasks as completed by setting the completed field to the current date using NOW(). It also ensures that only tasks that haven't been marked as completed yet using completed IS NULL are updated, preventing any unnecessary updates.



Saved: 5/2/2025 10:47:39 PM

100%

Task #3 (1 pt.) - Pending Page URLs

Url Prompt

Weight: 33.33%

Objective: Pending Page URLs

Details:

- Direct link to the file in the homework related branch from Github (should end in .php)
- Direct link to the file on Heroku Prod (Just grab the base prod url and manually enter the path to the file)

URL #1

<https://gs658-it202-008->

prod-96409305ed03.herokuapp.com/todos/pending.php



URL

<https://gs658-it202-008-prod-9640930>



URL #2

<https://github.com/goribantu/gs658->

IT202008M4-



URL

<https://github.com/goribantu/gs658-IT202008M4>



100%

Section #4: (2 pts.) Completed Page

100%

Task #1 (1 pt.) - Fetching Todos

Combo Task:

Weight: 50%

Objective: Fetching Todos

Image Prompt

Weight: 50%

Details:

- Refer to the HTML table below and build a query that'll select the columns in the same order as the table from the Todo table
 - Cross-reference the HTML table columns with what'd most plausibly match the SQL table aside from the notes below
 - For the completed date you'll need to extract the date portion from the completed column
 - For the Status part, you'll need to calculate the "days_offset" from the completed date, ensure the virtual column matches "days_offset"
 - Filter the results where the todo item is completed and order the results by most recently completed and most recently due
 - No limit is required
 - Include screenshot(s) from Heroku Dev demonstrating 5 different records (ensure URL is visible)

showing fetch solution



showing completed heroku output



Saved: 5/2/2025 10:52:07 PM

≡, Text Prompt

Weight: 50%

Details:

Briefly describe how the query works in plain words.

Your Response:

This query selects only the columns necessary for the completed Todos table. It also calculates the days_offset using DATEDIFF & allows the PHP to display whether a task was completed on time or overdue, and orders by completed and due so most recent tasks appear first.



Saved: 5/2/2025 10:52:07 PM

100%

Task #2 (1 pt.) - Completed Page URLs

⊕ Url Prompt

Weight: 50%

Objective: *Completed Page URLs*

Details:

- Direct link to the file in the homework related branch from Github (should end in .php)
- Direct link to the file on Heroku Prod (Just grab the base prod url and manually enter the path to the file)

URL #1

[https://gs658-it202-008-
prod-96409305ed03.herokuapp.com/todos/completed.php](https://gs658-it202-008-prod-96409305ed03.herokuapp.com/todos/completed.php)



URL

[https://gs658-it202-008-
prod-96409305ed03.herokuapp.com/todos/completed.php](https://gs658-it202-008-prod-96409305ed03.herokuapp.com/todos/completed.php)



URL #2

[https://github.com/goribau/gs658-
IT202-008M4-todos-completed.php](https://github.com/goribau/gs658-IT202-008M4-todos-completed.php)



URL

<https://github.com/goribau/gs658-IT202-008M4-todos-completed.php>



100%

Section #5: (1 pt.) Misc

100%

Task #1 (0.33 pts.) - Github Details

Combo Task:

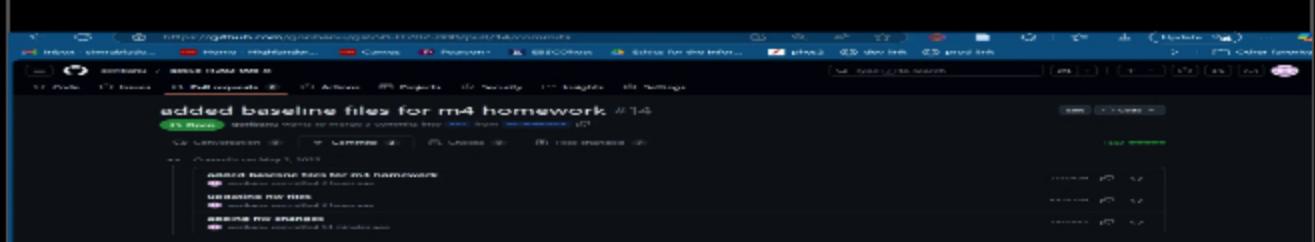
Weight: 33.33%

☒ Image Prompt

Weight: 60%

Details:

From the Commits tab of the Pull Request screenshot the commit history



commits



Saved: 5/2/2025 10:54:03 PM

☒ Url Prompt

Weight: 40%

Details:

Include the link to the Pull Request (should end in `/pull/#`)

URL #1

<https://github.com/goribantu/gs658/pull/14>



URL

<https://github.com/goribantu/gs658/pull/14>



Saved: 5/2/2025 10:54:03 PM

100%

Task #2 (0.33 pts.) - WakaTime - Activity

☒ Image Prompt

Weight: 33.33%

Objective: WakaTime - Activity

Details:

Details:

- Visit the WakaTime.com Dashboard
- Click **Projects** and find your repository
- Capture the overall time at the top that includes the repository name
- Capture the individual time at the bottom that includes the file time
- Note: The duration isn't relevant for the grade and the visual graphs aren't necessary

The screenshot shows the WakaTime dashboard for the project "gs658-IT202-008". At the top, it displays "2 hrs 37 mins over the Last 7 Days in gs658-IT202-008 under all branches." Below this, there's a large, mostly empty chart area. In the bottom left corner of the chart area, the text "wakatime activity" is visible, with a yellow arrow pointing down towards the detailed activity log below.

The screenshot shows a detailed activity log for the "gs658-IT202-008" project. The log lists numerous entries, each showing a file path and the duration spent on it. The log is very long and repetitive, showing many entries for "prakalita... /index.html" and other files like "prakalita... /index.html?view=tree&path=...". The total duration is listed as "2 hrs 37 mins" at the top right, and the branch is "main".

100%

Task #3 (0.33 pts.) - Reflection

Weight: 33.33%

Objective: *Reflection*

Sub-Tasks:

100%

Task #1 (0.33 pts.) - What did you learn?

Text Prompt

Weight: 33.33%

Objective: *What did you learn?*

Details:

Briefly answer the question (at least a few decent sentences)

Your Response:

I learned how to connect PHP to a MySQL database and actually writing the SQL queries that make things like adding and completing tasks work. I also learned how to handle form inputs, valid data, and even debugging when I kept getting errors when trying to submit the form (which happened a lot lol), but I got it to work at the end.



Saved: 5/2/2025 10:58:47 PM

100%

Task #2 (0.33 pts.) - What was the easiest part of the assignr

Text Prompt

Weight: 33.33%

Objective: *What was the easiest part of the assignment?*

Details:

Briefly answer the question (at least a few decent sentences)

Your Response:

The easiest part of the assignment was the completed.php and create.php stuff, like creating the tasks and adding them to the todo table. It was easy to kind of figure out the code for those types of things.



Saved: 5/2/2025 11:02:49 PM

100%

Task #3 (0.33 pts.) - What was the hardest part of the assignr

Text Prompt

Weight: 33.33%

Objective: *What was the hardest part of the assignment?*

Details:

Briefly answer the question (at least a few decent sentences)

Your Response:

The hardest part of this assignment for me was the pending.php stuff because I kept on getting errors when trying to add certain tasks to the table, so I had to go back and debug my code, double-check my SQL queries, and make sure the table names and column names matched correctly.

EDITION 2024 RELEASED, AND THERE ARE NO CHANGES AND NO NEW FEATURES IN THIS EDITION.



Saved: 5/2/2025 11:10:30 PM