Course: IT202-010-S2025
Assignment: IT202 Module 4 SQL Challenge
Student: Justin C. (jjc88)
Status: Submitted | Worksheet Progress: 100.00%
Potential Grade: 10.00/10.00 (100.00%)
Received Grade: 0.00/10.00 (0.00%)
Grading Link: [https://learn.ethereallab.app/assignment/v3/IT202-010-S2025/it202-module-4-sql-challenge/grading/jjc88](https://learn.ethereallab.app/assignment/v3/IT202-010-S2025/it202-module-4-sql-challenge/grading/jjc88)

# 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

7. Upload the same PDF to Canvas
8. Sync Local
    1. `git checkout dev`
    2. `git pull origin dev`

## Section #1: ( 1 pt.) Prep

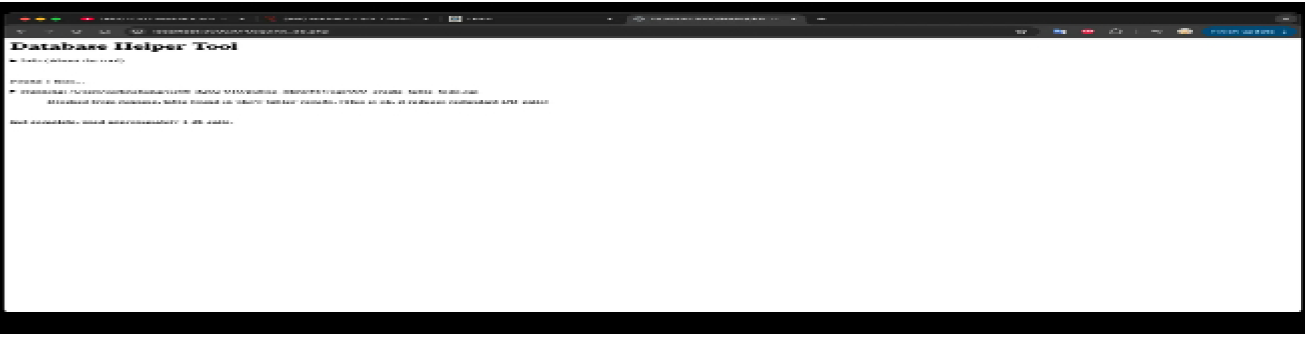
**Details:**
Once the files are copied in, run your local PHP server and visit /M4/sql/init_db.php in the browser to trigger the table generation.

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

**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)

### Image Prompt



Database Helper Tool

[illegible]

Output init_dp.php

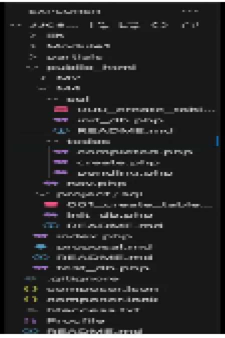
Saved: 3/11/2025 3:11:53 PM

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

**Weight:** *50%*

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

### Image Prompt



```
[illegible]
  > [illegible]
  > [illegible]
  > [illegible]
  v public_html
    > [illegible]
    v [illegible]
      v sql
        [illegible]
        init_db.php
        README.md
      v todos
        completed.php
        create.php
        pending.php
      nav.php
    v project / sql
      001_create_table...
      init_db.php
      README.md
    index.php
    [illegible].md
    README.md
    test_db.php
  .gitignore
  composer.json
  composer.lock
  [illegible].txt
  Procfile
  README.md
```

editor screenshot

Saved: 3/12/2025 4:57:27 PM

# Section #2: ( 3 pts.) Create Page

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

### Combo Task:

**Weight:** *25%*

**Objective:** *HTML Form Challenge*

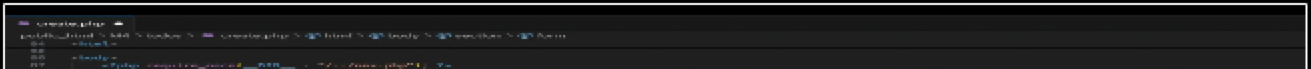
### Item:#1

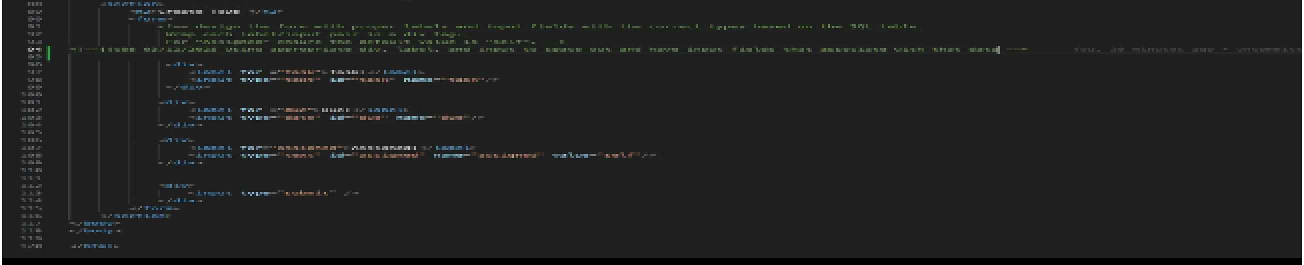
**Weight:** *50%*

**Details:**

Show the code snippet related to the completed HTML form

### Image Prompt





```
            <section>
                <h2>Create Task</h2>
                <form>
                    <!-- [illegible] design the form with proper labels and input fields with the correct types based on the [illegible] table
                     Wrap each label/input pair in a div tag.
                     For "assigned" ensure the default value is "self".  -->
    <!-- [illegible] [illegible] using appropriate div, label, and input to [illegible] out and have input fields that associate with that data -->

                     <div>
                        <label for ="task">Task</label>
                        <input type="text" id="task" name="task"/>
                     </div>

                    <div>
                        <label for ="due">Due</label>
                        <input type="date" id="due" name="due"/>
                    </div>

                    <div>
                        <label for="assigned">Assigned</label>
                        <input type="text" id="assigned" name="assigned" value="self"/>
                    </div>

                    <div>
                        <input type="submit" />
                    </div>
                </form>
            </section>
    </main>
    </body>

    </html>
```

HTML CODE

Saved: 3/12/2025 5:02:01 PM

### Item:#3

**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

### Text Prompt

Your Response:

For task, text was chosen as the task will be a string, due date will be a date type insuring only a date format, and assigned will be text as that is a string input. Assigned was able to be defaulted to self by assigning the value to "self" making the field input defaulted to self.

Saved: 3/12/2025 5:02:01 PM

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

### Combo Task:

**Weight:** *25%*
**Objective:** *Form Submission PHP Validations*

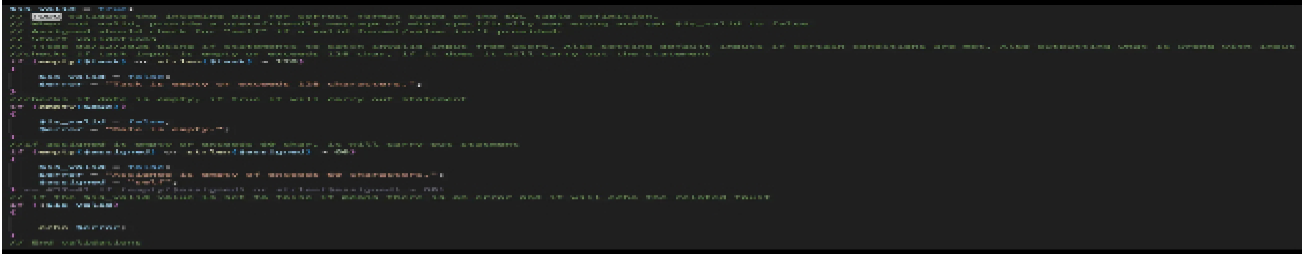
### Item:#1

**Weight:** *50%*

**Details:**

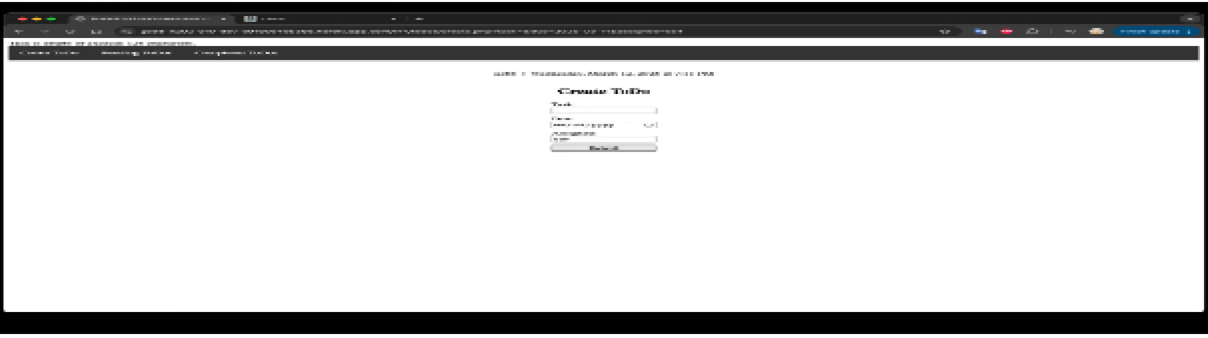
- Validate the incoming data for correct format based on the SQL table definition.
- When not valid, provide a user-friendly message of what specifically was wrong and set `$is_valid` to false
- Assigned should check for "self" if a valid format/value isn't provided
- Include screenshot(s) from Heroku Dev demonstrating each validation (ensure URL is visible)

## Image Prompt



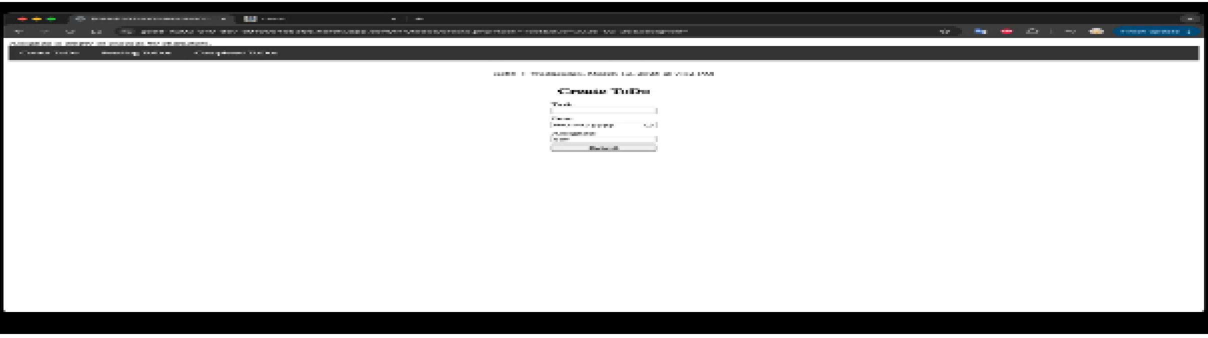
```
$is_valid = true;
// TODO validate the incoming data for correct format based on the SQL table definition.
// When not valid, provide a user-friendly message of what specifically was wrong and set $is_valid to false
// Assigned should check for "self" if a valid format/value isn't provided
// Start validations
// [illegible]
//checks if task input is empty or exceeds 130 char, if it does it will carry out the statement
if (empty($task) || strlen($task) > 130)
{
    $is_valid = false;
    $error = "Task is empty or exceeds 130 characters.";
}
//checks if date is empty, if true it will carry out statement
if (empty($date))
{
    $is_valid = false;
    $error = "Date is empty.";
}
//if assigned is empty or exceeds 60 char, it will carry out statement
if (empty($assigned) || strlen($assigned) > 60)
{
    $is_valid = false;
    $error = "Assigned is empty or exceeds 60 characters.";
    $assigned = "self";
} [illegible]
// if the $is_valid value is set to false it means there is an error and it will echo the related fault
if (!$is_valid)
{

    echo $error;
}
// End validations
```

code



Create ToDo

task



Create ToDo

assigned validation

Create ToDo

date validatiion

Saved: 3/12/2025 7:19:16 PM

## Item:#3

**Weight:** *50%*

**Details:**
Briefly explain your validation logic.

### Text Prompt

Your Response:

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

## Combo Task:

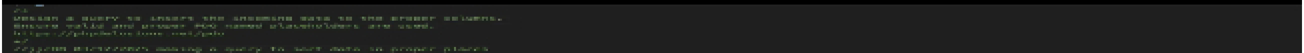
**Weight:** *25%*
**Objective:** *Show the code snippet related to the INSERT query*

## Item:#1

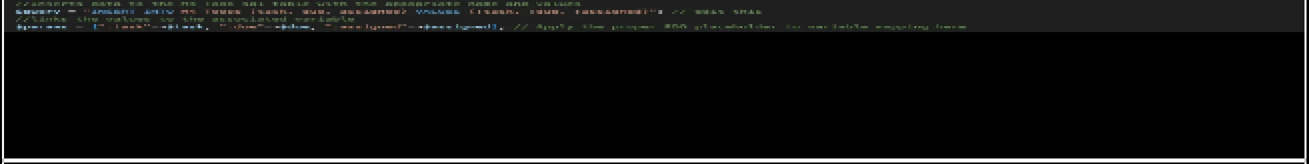
**Weight:** *50%*

**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

### Image Prompt



```
/*
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
*/
//[illegible] 3/12/2025 making a query to sort data in proper places
```



```
//inserts data to the M4_Todo sql table with the appropriate name and values
$query = "INSERT INTO M4_Todos (task, due, assigned) VALUES (:task, :due, :assigned)"; // edit this
//links the values to the associated variable
$params = [":task"=>$task, ":due"=>$due, ":assigned"=>$assigned]; // Apply the proper PDO placeholder to variable mapping here
```

query code

Saved: 3/12/2025 5:08:07 PM

## Item:#3

**Weight:** *50%*

**Details:**
Briefly describe `how` the query works in plain words.

### Text Prompt

Your Response:

Inserts data to the M4_Todo sql table with the appopriate name and values. Parms also links the values to the associate variable.

Saved: 3/12/2025 5:08:07 PM

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

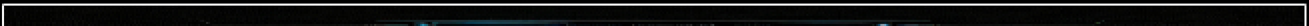
## Combo Task:

**Weight:** *31.25%*
**Objective:** *Extra Credit: Handle Unique Constraint Errors*

## Item:#1

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

### Image Prompt



Missing Caption

Not saved yet

## Item:#3

**Details:**
Briefly explain `how` your code solves the challenge.

### Text Prompt

Your Response:

Missing Response

Not saved yet

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

**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 Prompt

URL #1
https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com//M4/todos/create.php

URL
https://jjc88-it202-010-prod-b44e1de4(

URL #2

URL
https://github.com/juicejoose/jjc88-it2

https://github.com/juicejoose/jjc88-it202-101/M4-Homework/public_html/M4/todos/create.php

https://github.com/juicejoose/jjc88-it2

Saved: 3/12/2025 8:16:01 PM

# Section #3: ( 3 pts.) Pending Page

## Task #1 ( 1 pt.) - Fetching ToDos

### Combo Task:

**Weight:** *33.33%*
**Objective:** *Fetching ToDos*

### Item:#1

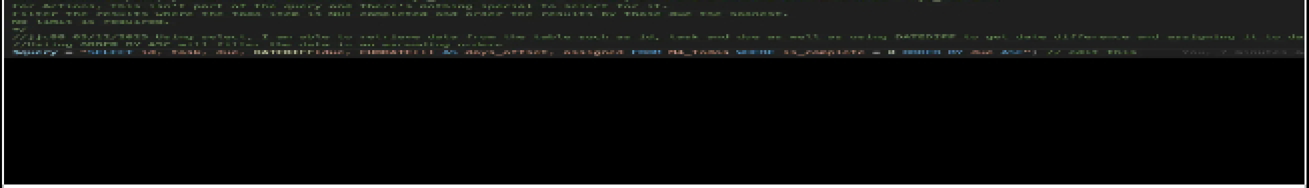
**Weight:** *50%*

**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)

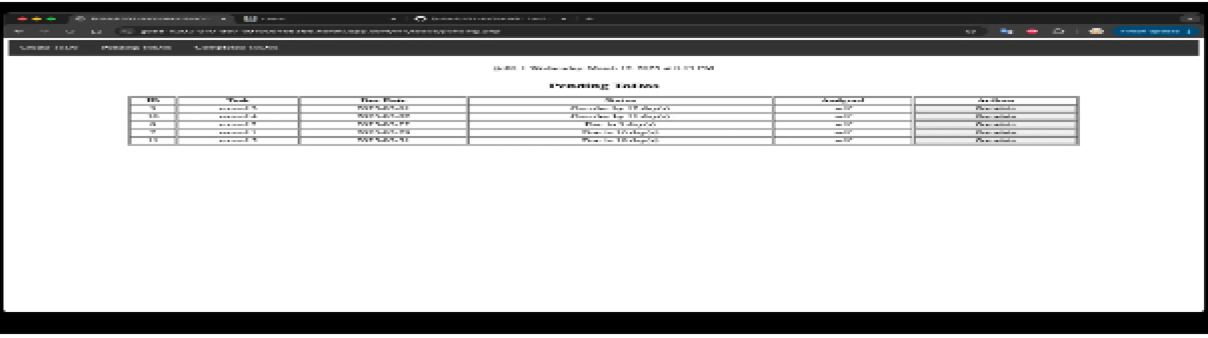
#### Image Prompt

```
/* 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.
Crossreference the HTML table columns with what'd most plausibly match the SQL table aside from the notes below.
```



```
for actions, this isn't part of the query and there's nothing special to select for it.
Filter the results where the task item is not completed and order the results by those due the soonest.
No limit is required.
*/
//jj.88 03/12/2025 Using select, I am able to retrieve data from the table such as id, task and due as well as using DATEDIFF to get date difference and assigning it to da
//Using ORDER BY ASC will filter the data in an ascending order.
$query = "SELECT id, task, due, DATEDIFF(due, CURDATE()) AS days_offset, assigned FROM M4_Tasks WHERE is_completed = 0 ORDER BY due ASC"; // edit this
```

code



Pending Tasks

| ID | Task | Due Date | Status | Assigned | Actions |
|---|---|---|---|---|---|
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

Showing 5 records

Saved: 3/12/2025 8:21:28 PM

## Item:#3

**Weight:** *50%*

**Details:**
Briefly describe `how` the query works in plain words.

### Text Prompt

Your Response:

Using select, I am able to retrieve data from the table such as id, task and due as well as using DATEDIFF to get date difference and assigning it to days_offset from table. Using ORDER BY ASC will filter the data in an ascending order.

Saved: 3/12/2025 8:21:28 PM

## Task #2 ( 1 pt.) - Completing ToDos

### Combo Task:

**Weight:** *33.33%*
**Objective:** *Completing ToDos*

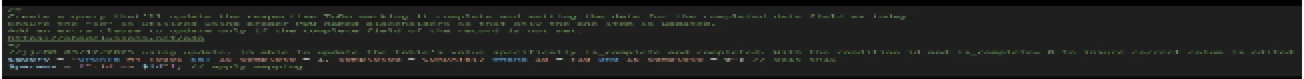
## Item:#1

**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

### Image Prompt



```
/*
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.
[illegible]
*/
//[illegible] 03/12/2025 using update, im able to update the table's value specifically is_complete and completed. With the condition id and is_complete= 0 to insure correct colum is edited
$query = "[illegible]"; // [illegible]
$params = [":id" => $id]; // apply mapping
```

code

Saved: 3/12/2025 6:01:42 PM

## Item:#3

**Weight:** *50%*

**Details:**
Briefly describe `how` the query works in plain words.

### Text Prompt

Your Response:

using update, im able to update the table's value specifically is_complete and completed. With the condition id and is_complete= 0 to insure correct colum is edited.

Saved: 3/12/2025 6:01:42 PM

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

**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 Prompt

URL #1
https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com//M4/todos/pending.php

URL
https://jjc88-it202-010-prod-b44e1de4

URL #2
https://github.com/juicejoose/jjc88-it202-010/blob/M4-Homework/public_html/M4/todos/pending.php

URL
https://github.com/juicejoose/jjc88-it2

Saved: 3/12/2025 8:22:20 PM

# Section #4: ( 2 pts.) Completed Page

## Task #1 ( 1 pt.) - Fetching ToDos

### Combo Task:

**Weight:** *50%*
**Objective:** *Fetching ToDos*

### Item:#1

**Weight:** *50%*

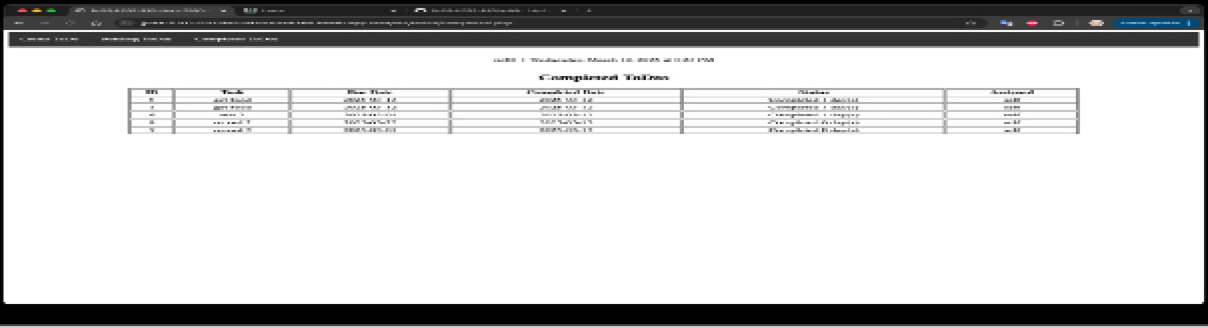
**Details:**

- 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)

## Image Prompt

```
[illegible]
```

code



Completed ToDos

| ID | Task | Due Date | Completed Date | Status | Assigned |
|---|---|---|---|---|---|
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

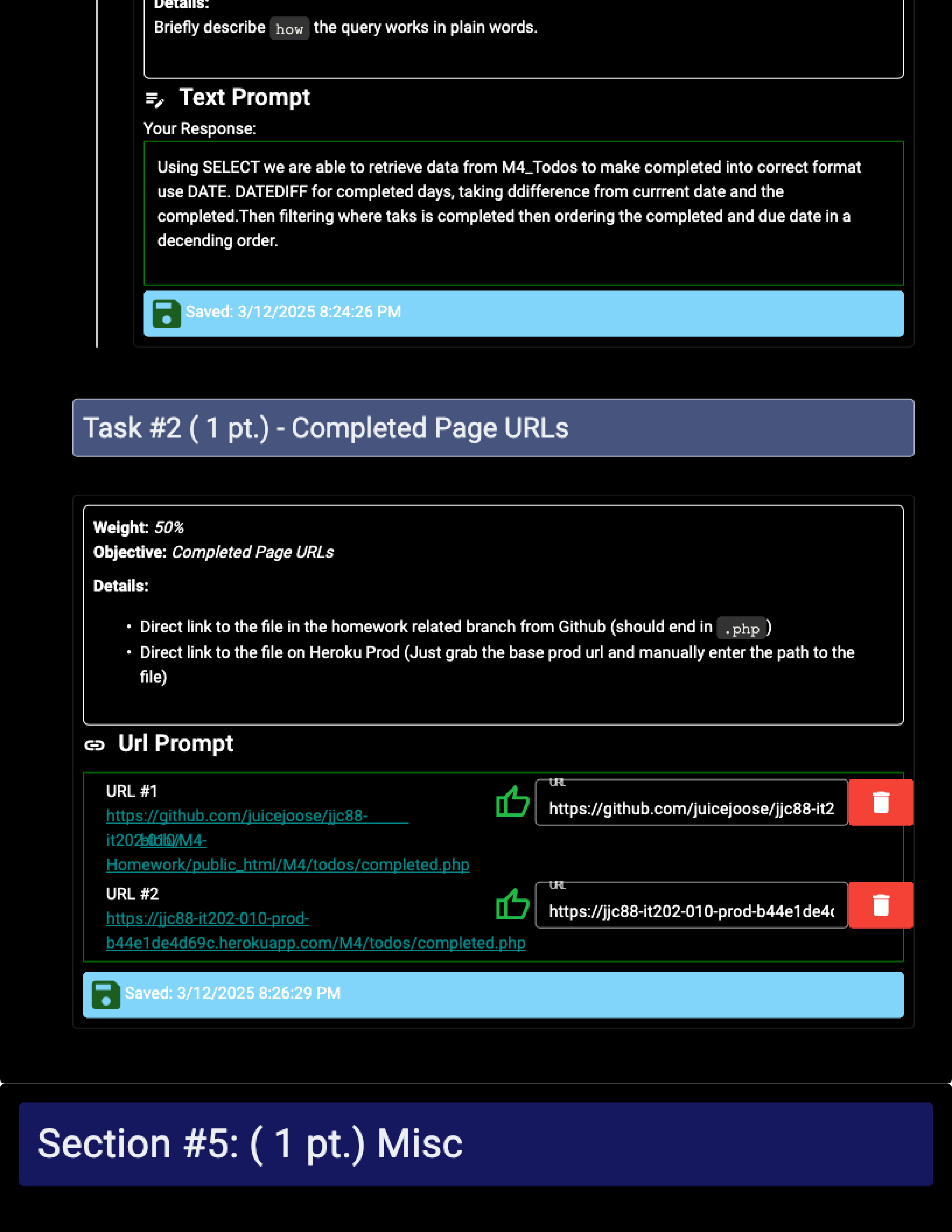
heroku 5 records

Saved: 3/12/2025 8:24:26 PM

## Item:#3

**Weight:** *50%*

**Details:**



**Details:**
Briefly describe `how` the query works in plain words.

### Text Prompt

Your Response:

Using SELECT we are able to retrieve data from M4_Todos to make completed into correct format use DATE. DATEDIFF for completed days, taking ddifference from currrent date and the completed.Then filtering where taks is completed then ordering the completed and due date in a decending order.

Saved: 3/12/2025 8:24:26 PM

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

**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 Prompt

URL #1
https://github.com/juicejoose/jjc88-it202-010/blob/M4-Homework/public_html/M4/todos/completed.php

URL
https://github.com/juicejoose/jjc88-it2

URL #2
https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com/M4/todos/completed.php

URL
https://jjc88-it202-010-prod-b44e1de4

Saved: 3/12/2025 8:26:29 PM

# Section #5: ( 1 pt.) Misc

# Task #1 ( 0.33 pts.) - Github Details

## Combo Task:

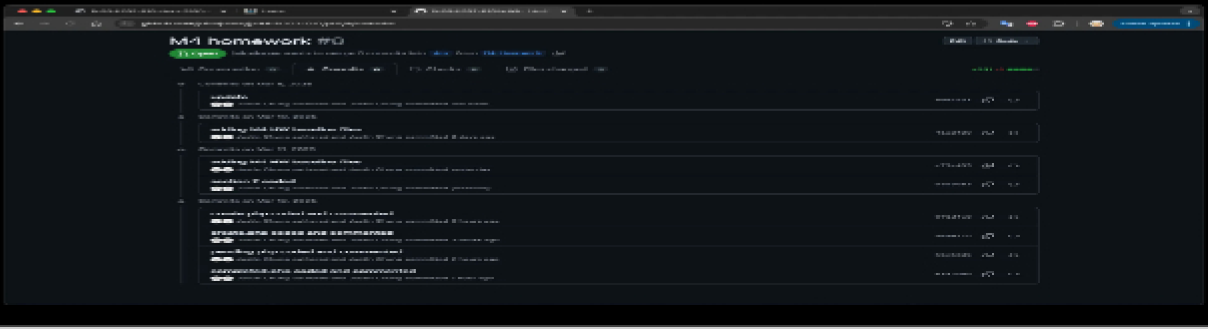
**Weight:** *33.33%*
**Objective:** *Github Details*

### Item:#1

**Weight:** *60%*

**Details:**
From the Commits tab of the Pull Request screenshot the commit history

### Image Prompt



M4 homework #0

github commit

Saved: 3/12/2025 8:29:22 PM

### Item:#2

**Weight:** *40%*

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

### Url Prompt

URL #1
https://github.com/juicejoose/jjc88-it202pull/08

URL
https://github.com/juicejoose/jjc88

Saved: 3/12/2025 8:29:22 PM

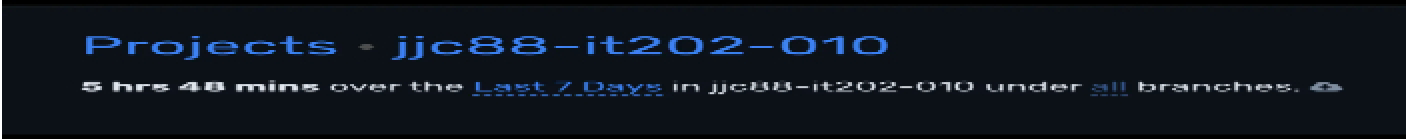
## Task #2 ( 0.33 pts.) - WakaTime - Activity

**Weight:** *33.33%*
**Objective:** *WakaTime - Activity*

**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

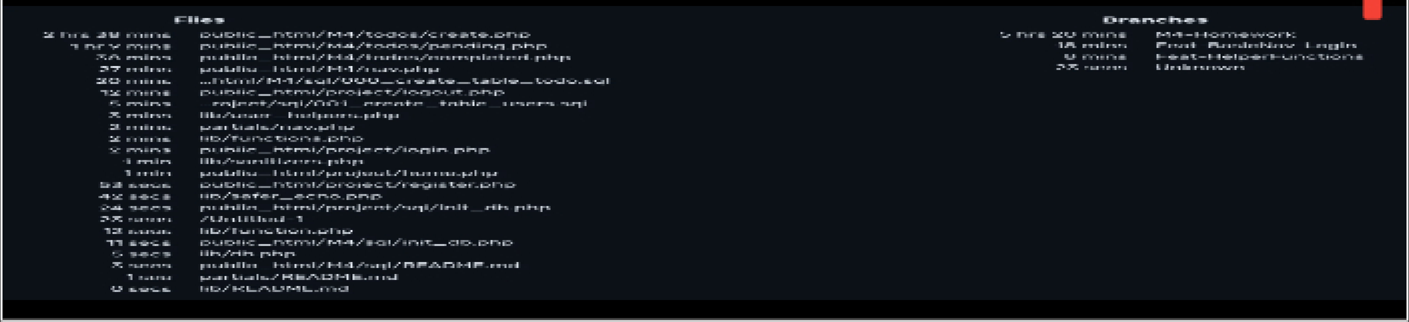
### Image Prompt



Projects • jjc88-it202-010

**5 hrs 48 mins** over the Last 7 Days in jjc88-it202-010 under all branches.

wakatime summary



```
                    Files
2 hrs 28 mins    public_html/M4/todos/create.php
  1 hr 7 mins    public_html/M4/todos/pending.php
      36 mins    public_html/M4/todos/completed.php
      27 mins    public_html/M4/nav.php
      20 mins    ...html/M4/sql/000_create_table_todo.sql
      12 mins    public_html/project/logout.php
       6 mins    ...roject/sql/001_create_table_users.sql
       3 mins    lib/user_helpers.php
       3 mins    partials/nav.php
       2 mins    lib/functions.php
       2 mins    public_html/project/login.php
        1 min    lib/sanitizers.php
        1 min    public_html/project/home.php
      53 secs    public_html/project/register.php
      42 secs    lib/safer_echo.php
      24 secs    public_html/project/sql/init_db.php
      23 secs    /Untitled-1
      12 secs    lib/function.php
      11 secs    public_html/M4/sql/init_db.php
       5 secs    lib/db.php
       3 secs    public_html/M4/sql/README.md
        1 sec    partials/README.md
       0 secs    lib/README.md
```

```
                    Branches
5 hrs 20 mins    M4-Homework
      18 mins    Feat-BasicNav-Login
       9 mins    Feat-HelperFunctions
      23 secs    Unknown
```

## Task #3 ( 0.33 pts.) - Reflection

### Sub-Tasks:

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

**Weight:** *33.33%*
**Objective:** *What did you learn?*

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

### Text Prompt

Your Response:

Setting up pages using php, html, and sql. A combination of everythiing working together to make a page wittth funcattionality. Also learning how to build queries, setting up validations as well as building using html.

Saved: 3/12/2025 8:35:40 PM

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

**Weight:** *33.33%*
**Objective:** *What was the easiest part of the assignment?*

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

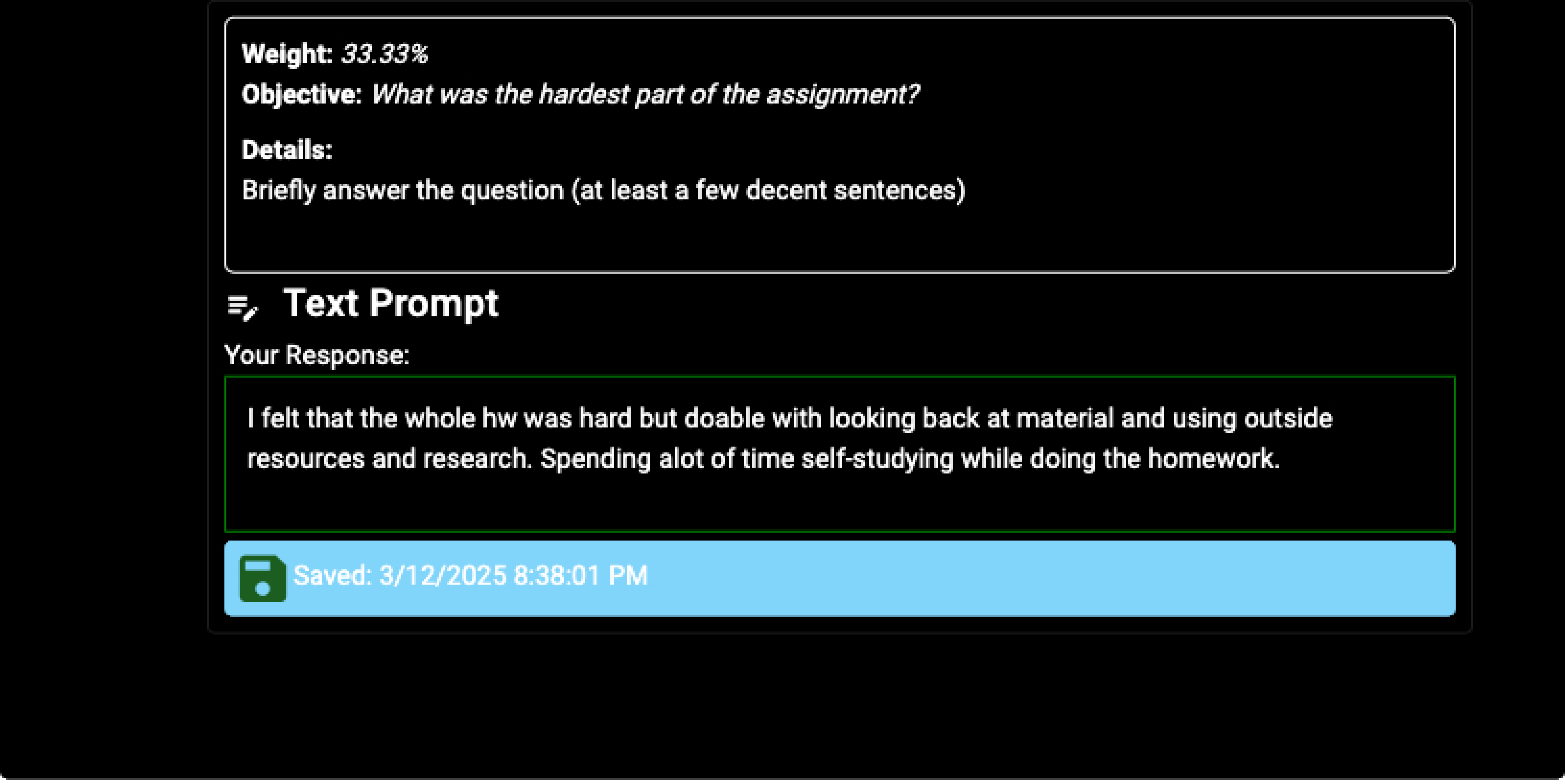
### Text Prompt

Your Response:

Easiest part was submitting even though my laptop is gonna explode from all the heat and lag. But I wouldn't say any easy part of the hw every part had its challenges.

Saved: 3/12/2025 8:37:01 PM

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



**Weight:** *33.33%*

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

**Details:**

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

### Text Prompt

Your Response:

I felt that the whole hw was hard but doable with looking back at material and using outside resources and research. Spending alot of time self-studying while doing the homework.

Saved: 3/12/2025 8:38:01 PM