

Course: IT202-008-S2025

Assignment: IT202 Module 4 SQL Challenge

Student: Dylan W. (dw347)

Status: Submitted | Worksheet Progress: 105.88%

Potential Grade: 10.90/10.00 (109.00%)

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/dw347>

# 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

## 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

Running: /app/public\_html/M4/sql/000\_create\_table\_todo.sql

Blocked from running; table found in 'dw347-it202-000-dev-f023cc1dab09.herokuapp.com/M4/sql/init\_db.php'

init\_db.php blocked

Saved: 3/8/2025 11:45:08 AM

### 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



## showing files in vscode



Saved: 3/8/2025 11:49:42 AM

## 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

```
task = text type and it ensures a max character lenght isn't exceeded due = data value which uses a date input allowing dates to be inputed assigned = text type which stores the name of the person assigned to the task I made self the default by using the value attribute. that prefills the input field when the form loads. self will always be there if not changed.
```

showing form field



Saved: 3/8/2025 12:06:16 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:

task = text type and it ensures a max character lenght isn't exceeded due = data value which uses a date input allowing dates to be inputed assigned = text type which stores the name of the person assigned to the task I made self the default by using the value attribute. that prefills the input field when the form loads. self will always be there if not changed.



Saved: 3/8/2025 12:06:16 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

```
26 validate = array_diff($keynames, array_keys($values));
27
28 if (empty($values)) {
29     // User submitted, must be valid
30     $task = trim($this->task);
31     $value = trim($values['value']); // Input must be a valid MySQL date format
32     $deserialized = trim($values['deserialized']); // Must be 'true' or a valid format (not empty or equivalent)
33
34     $is_valid = true;
35     $errors = [];
36
37     // Create validation
38     if (empty($task)) {
39         $is_valid = false;
40         $errors[] = "Task cannot be empty.";
41     }
42
43     if (!preg_match('/^(\d{4})-(\d{2})-(\d{2}) (\d{2}):(\d{2}):(\d{2})$/i', $value) || !is_numeric($value)) {
44         $is_valid = false;
45         $errors[] = "Date must be a valid date in YYYY-MM-DD format.";
```



 Saved: 3/8/2025 12:21:22 PM

### Item:#3

**Weight: 50%**

#### **Details:**

**Briefly explain your validation logic.**

## ≡, Text Prompt

### Your Response:

The code ensures the task input isn't empty, it also ensures that the date follows a certain format.

The code ensures the task input isn't empty. It also ensures that the date follows a certain format. It also ensures that the assigned to isn't empty ensuring that self is the default input. If the is\_valid is true then the code proceeds, and it collects errors messages if validation fails.



Saved: 3/8/2025 12:21:22 PM

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

### 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://phodelusions.net/pdo>

### ≡, Image Prompt

```
git@phoebus:~/Dev/2025/03/08/insert-pdo$ ./index.php
14  if ($is_valid) {
15      // Design a query to insert the incoming data to the proper columns
16      // Ensure valid and proper PDO named placeholders are used
17      $query = "INSERT INTO Tasks (task_id, due, assigned, VALUE) VALUES (:task_id, :due, :assigned, :value)";
18      $stmt = $pdo->prepare($query);
19
20      $stmt->bindParam(':task_id', $task_id);
21      $stmt->bindParam(':due', $due);
22      $stmt->bindParam(':assigned', $assigned);
23      $stmt->bindParam(':value', $value);
24
25      $stmt->execute();
26
27      // Check if an exception was related to a unique constraint
28      if ($stmt->errorCode() == 23000) {
29          echo "Duplicate entry error. This task already exists for the specified due date." . PHP_EOL;
30      } else {
31          echo "An error occurred inserting the record. Check the logs for details." . PHP_EOL;
32          error_log("INSERTING($query) - [" . var_export($stmt, true) . "] // Error: " . $stmt->errorInfo());
33      }
34  } else {
35      echo "The task input is empty." . PHP_EOL;
36      echo "The due date is invalid." . PHP_EOL;
37      echo "The assigned user is empty." . PHP_EOL;
38  }
}
showing insert query
```



Saved: 3/8/2025 12:35:56 PM

### Item:#3

**Weight:** 50%

**Details:**

Details.

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

## ≡, Text Prompt

Your Response:

user submits a to do query form with the task, due date and assigned. for data must be validated and if it fails, it shows error messages. if it passes, it proceeds to the database insertion. sql insert is also executed using PDO with bound parameters. if successful it will show the inserted tasks ID and if it fails, is will show an error message and takes note of the issue.



Saved: 3/8/2025 12:35:56 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

```
1 // echo "Failed to insert!";
2 // echo "Description: $desc";
3 // echo "The insertion was rejected by a unique constraint";
4 // if ($errorCode == 23000) {
5 //   echo "Duplicate entry error: this task already exists for the specified due date.";
6 // } else {
7 //   echo "There was an error inserting the record which the value: '$desc' ";
8 //   echo "was inserted. Below are the commands used to reinsert the record: ";
9 //   echo "-----";
10 //   echo "use $db; ";
11 //   echo "insert into tasks (due_date, description) values ('$dueDate', '$desc') ";
12 //   echo "on duplicate key update description = '$desc' ";
13 // }
14 // echo "-----";
```

solution for extra credit



Saved: 3/8/2025 12:59:26 PM

## Item:#3

### Details:

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

### Text Prompt

Your Response:

php validates the user input and inserts a new to do task into teh databse using PDO. first it validates the input then prepares the insert sql query using pdo. PDO prepared statements prevent sql injection. it also checks the error code 23000 whcih occurs when a task with the same name and due date already exists. if everything is good, it displays the inserted tasks ID.



Saved: 3/8/2025 12:59:26 PM

## 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://dw347-it202-008->

<prod-39957dda50b0.herokuapp.com/M4/todos/create.php>



UR

<https://dw347-it202-008-prod-39957dd>



URL #2

<https://github.com/dyylan2018/dw347->

<IT202008M4->

[Homework/public\\_html/M4/todos/create.php](Homework/public_html/M4/todos/create.php)



UR

<https://github.com/dyylan2018/dw347>



Saved: 3/8/2025 1:02:21 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

## showing fetch code

Pending ToDos						
#	Task	Due Date	Status	Assigned	Assigned	Priority
1	Do HW	2023-08-10	Due in 2 days(s)	self	self	Normal
2	Reading	2023-08-10	Due in 2 days(s)	self	self	Normal
3	Finish PPT slides	2023-08-10	Due in 2 days(s)	self	self	Normal
4	Reading PPT	2023-08-10	Due in 2 days(s)	self	self	Normal
5	Spring break	2023-08-10	Due in 6 days(s)	self	self	Normal

dev output



Saved: 3/8/2025 1:17:25 PM

## Item:#3

**Weight:** 50%

**Details:**

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

### Text Prompt

Your Response:

the code selects specific columns then analyzes them. it also filters the pending tasks then sorts them by due date. overall the code runs the query, it fetches the results and handles errors.



Saved: 3/8/2025 1:17:25 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 completion.

- Create a query that will 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

```
1 // process command line arguments
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...
101 // ...
102 // ...
103 // ...
104 // ...
105 // ...
106 // ...
107 // ...
108 // ...
109 // ...
110 // ...
111 // ...
112 // ...
113 // ...
114 // ...
115 // ...
116 // ...
117 // ...
118 // ...
119 // ...
120 // ...
121 // ...
122 // ...
123 // ...
124 // ...
125 // ...
126 // ...
127 // ...
128 // ...
129 // ...
130 // ...
131 // ...
132 // ...
133 // ...
134 // ...
135 // ...
136 // ...
137 // ...
138 // ...
139 // ...
140 // ...
141 // ...
142 // ...
143 // ...
144 // ...
145 // ...
146 // ...
147 // ...
148 // ...
149 // ...
150 // ...
151 // ...
152 // ...
153 // ...
154 // ...
155 // ...
156 // ...
157 // ...
158 // ...
159 // ...
160 // ...
161 // ...
162 // ...
163 // ...
164 // ...
165 // ...
166 // ...
167 // ...
168 // ...
169 // ...
170 // ...
171 // ...
172 // ...
173 // ...
174 // ...
175 // ...
176 // ...
177 // ...
178 // ...
179 // ...
180 // ...
181 // ...
182 // ...
183 // ...
184 // ...
185 // ...
186 // ...
187 // ...
188 // ...
189 // ...
190 // ...
191 // ...
192 // ...
193 // ...
194 // ...
195 // ...
196 // ...
197 // ...
198 // ...
199 // ...
200 // ...
201 // ...
202 // ...
203 // ...
204 // ...
205 // ...
206 // ...
207 // ...
208 // ...
209 // ...
210 // ...
211 // ...
212 // ...
213 // ...
214 // ...
215 // ...
216 // ...
217 // ...
218 // ...
219 // ...
220 // ...
221 // ...
222 // ...
223 // ...
224 // ...
225 // ...
226 // ...
227 // ...
228 // ...
229 // ...
230 // ...
231 // ...
232 // ...
233 // ...
234 // ...
235 // ...
236 // ...
237 // ...
238 // ...
239 // ...
240 // ...
241 // ...
242 // ...
243 // ...
244 // ...
245 // ...
246 // ...
247 // ...
248 // ...
249 // ...
250 // ...
251 // ...
252 // ...
253 // ...
254 // ...
255 // ...
256 // ...
257 // ...
258 // ...
259 // ...
260 // ...
261 // ...
262 // ...
263 // ...
264 // ...
265 // ...
266 // ...
267 // ...
268 // ...
269 // ...
270 // ...
271 // ...
272 // ...
273 // ...
274 // ...
275 // ...
276 // ...
277 // ...
278 // ...
279 // ...
280 // ...
281 // ...
282 // ...
283 // ...
284 // ...
285 // ...
286 // ...
287 // ...
288 // ...
289 // ...
290 // ...
```

how todos are completed



Saved: 3/8/2025 1:21:22 PM

## Item:#3

**Weight:** 50%

**Details:**

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

## ≡, Text Prompt

Your Response:

the code checks the user to see if they submitted a task. it marks the task as complete and sets completion date. it also ensures that incomplete tasks are updated. it also includes PDO queries to prevent sql injections.



Saved: 3/8/2025 1:21:22 PM

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

**Weight:** 33.33%

**Objective:** Pending Page URLs

**Details:**

### 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://dw347-it202-008->



URL

<https://dw347-it202-008-prod-39957dd>



URL #2

<https://github.com/dyylan2018/dw347->



URL

<https://github.com/dyylan2018/dw347->



[IT202-H008M4-Homework/public\\_html/M4/todos/pending.php](IT202-H008M4-Homework/public_html/M4/todos/pending.php)



Saved: 3/8/2025 1:22:01 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 `matchas_days_offset`

- 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

fetch solutions

ID	Title	Due Date	Completed Date	Status	Assigned
1	Upgrading hardware	2022-03-01	2022-03-01	Completed on time	user1
2	Setting up new server	2022-03-02	2022-03-02	Completed on time	user1
3	Finalize IT-PPD Requests	2022-03-03	2022-03-03	Completed on time	user1
4	Test new server	2022-03-04	2022-03-04	Completed on time	user1
5	Training	2022-03-05	2022-03-05	Completed on time	user1

heroku dev output

 Saved: 3/8/2025 1:39:04 PM

### Item:#3

**Weight: 50%**

#### **Details:**

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

## Text Prompt

Your Response:

this code fetches completed tasks from the todos table. it calculates how many days late or early the task was. it sorts tsks by completion day. it sorts the results in an array and handles errors by displaying a message.



Saved: 3/8/2025 1:39:04 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/dyylan2018/dw347-IT202-008-M4-Homework/public\\_html/M4/todos/completed.php](https://github.com/dyylan2018/dw347-IT202-008-M4-Homework/public_html/M4/todos/completed.php)



tt

[https://github.com/dyylan2018/dw347-IT202-008-M4-Homework/public\\_html/M4/todos/completed.php](https://github.com/dyylan2018/dw347-IT202-008-M4-Homework/public_html/M4/todos/completed.php)



URL #2

<https://dw347-it202-008-prod-39957dd.herokuapp.com/M4/todos/completed.php>



tt

<https://dw347-it202-008-prod-39957dd.herokuapp.com/M4/todos/completed.php>



Saved: 3/8/2025 1:25:01 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

A screenshot of a GitHub pull request interface. The title of the pull request is "solved todos #38". The commit history shows three commits:

- dyylan2018 committed 3 hours ago: Add todo list to header
- dyylan2018 merged common feature into dw347: Merge pull request from dyylan2018 into dw347
- dyylan2018 committed 3 hours ago: Implementing the program to display a list of todos with checkboxes

The right sidebar shows the following statistics:

- Branch: main
- File changes: 2
- Lines added: 12
- Lines removed: 12
- Commit: 3
- Author: dyylan2018
- Reviewers: 0
- Labels: 0
- Mergeable: No

did them separately



A screenshot of a GitHub pull request interface. The title of the pull request is "added baseline files for M4 HW #36". The commit history shows two commits:

- dyylan2018 committed 1 day ago: Add baseline files for M4 HW #36
- dyylan2018 merged common feature into dw347: Merge pull request from dyylan2018 into dw347

The right sidebar shows the following statistics:

- Branch: main
- File changes: 2
- Lines added: 12
- Lines removed: 12
- Commit: 2
- Author: dyylan2018
- Reviewers: 0
- Labels: 0
- Mergeable: Yes

did them separately



Saved: 3/8/2025 1:43:08 PM

## Item:#2

**Weight:** 40%

**Details:**

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

### Url Prompt

URL #1

<https://github.com/dylan2018/dw347-IT2020036>



URL

<https://github.com/dylan2018/dw347-IT2020036>



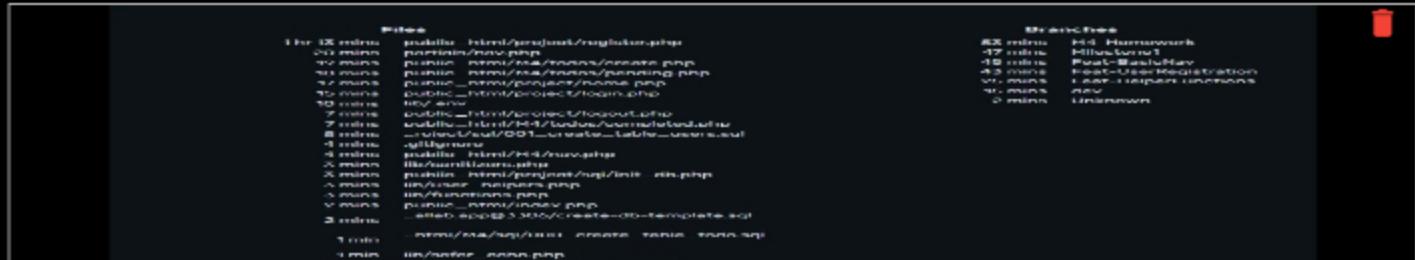
Task #2 ( 0.33 pts.) - WakaTime - Activity

**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



### 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:

I learned how to call a certain variable or type to stay default for an awaiting input or none at all. I learned how to implement sql queries into data and send error messages if input doesn't match necessary credentials. assign value types and data values along with other stuff.



Saved: 3/8/2025 1:50:06 PM

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

**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:

making the self type default was pretty easy for me



Saved: 3/8/2025 1:50:51 PM

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

**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:

The extra credit was pretty difficult but i figured it out in the end.



Saved: 3/8/2025 1:50:33 PM