

Submission Worksheet

Submission Data

Course: IT202-450-M2025

Assignment: IT202 Milestone 2

Student: Nathanael G. (ng569)

Status: Submitted | **Worksheet Progress:** 100%

Potential Grade: 10.00/10.00 (100.00%)

Received Grade: 0.00/10.00 (0.00%)

Started: 7/25/2025 5:10:45 PM

Updated: 7/25/2025 8:08:59 PM

Grading Link: <https://learn.ethereallab.app/assignment/v3/IT202-450-M2025/it202-milestone-2/grading/ng569>

View Link: <https://learn.ethereallab.app/assignment/v3/IT202-450-M2025/it202-milestone-2/view/ng569>

Instructions

- Overview Link: https://youtu.be/t6X8dzZt_a0

1. Refer to Milestone2 of this doc:

<https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88EWVwfo/view>

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

3. Ensure you've gone through each lesson related to this Milestone

4. Switch to the Milestone2 branch

1. git checkout Milestone2 (ensure proper starting branch)
2. git pull origin Milestone2 (ensure history is up to date)

5. Fill out the below worksheet

- Ensure there's a comment with your UCID, date, and brief summary of the snippet in each screenshot
- Ensure proper styling is applied to each page
- Ensure there are no visible technical errors; only user-friendly messages are allowed

6. Once finished, click "Submit and Export"

7. 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 Milestone2
4. On Github merge the pull request from Milestone2 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)

8. Upload the same PDF to Canvas

9. Sync Local

10. git checkout dev

11. git pull origin dev

Section #1: (2 pts.) Data

Progress: 100%

≡ Task #1 (0.50 pts) - API Data Table

Part 1:

Details:

- Show the table(s) structure of the table made to hold your API
- There should be at least 3 properties/fields beyond `id`, `created`, `modified`, and respective `is_api` or `api_id` columns
- Logical types and constraints should be set
- Show at least a few records from each API table (samples should include both API fetched data and custom entries noted by an `api_id` or `is_api` column)

	<code>id</code>	<code>email</code>	<code>password</code>	<code>created</code>	<code>modified</code>	<code>username</code>	<code>location</code>	<code>tz_name</code>	<code>tz_loc</code>	<code>tz_abbr</code>	<code>gmt</code>
>	1	ng569@mit.edu	\$2y\$12\$pq9gG0HLC7t9N	2025-06-20 02:41:24	2025-07-23 08:42:21	ng569	New York	Eastern Daylight Time	America/New_York	EDT	-4
>	4	demo@test.com	\$2y\$12\$6deU42l951Pq6R	2025-06-20 01:23:50	2025-07-22 06:05:53	deman	New York	Eastern Daylight Time	America/New_York	EDT	-4
>	6	test@test.com	\$2y\$12\$unbxnJfWknFy3q	2025-07-07 06:24:45	2025-07-22 06:05:53	test	New York	Eastern Daylight Time	America/New_York	EDT	-4
>	7	test@test.com	\$2y\$12\$VshvAD7mIuSj	2025-07-25 03:46:37	2025-07-25 03:46:37	mathg	London	British Summer Time	Europe/London	BST	1

VS Code SQL API data table (mixed with Users table)



Saved: 7/25/2025 5:16:11 PM

Part 2:

Details:

- Add a direct link to your sql file from Github (should end in `.sql`)

URL #1

<https://github.com/Nate-Gaw/ng569->

URL

<https://github.com/Nate-Gaw/ng569->

IT202450Milestone2/public_html/project/sql/011_insert_api_columns.sql



Saved: 7/25/2025 5:16:11 PM

Part 3:

Details:

- Briefly note what each field/property represents

Your Response.

So the API is used as a verification. When the user registers or changes location on their profile, the input is put into the API. If the api doesn't return anything, then the location doesn't exist. If it does, it's then parsed into these columns for each person. This allows for time to be accurately taken and used. The tz_name is the name of the timezone of where the user is so Eastern Daylight Time, for example. Then tz_loc is the location of where the user actually is (ex. America/New_York). Tz_abb is the abbreviation given by the API of the timezone (ex. EDT). gmt is the gmt offset time from the api (ex. -4)



Saved: 7/25/2025 5:16:11 PM

☰ Task #2 (0.50 pts.) - Data Transformation

Progress: 100%

▣ Part 1:

Progress: 100%

Details:

- Show the code that handles the data transformation/conversion (from the API)
- Show a Heroku dev example of API getting fetched and loaded into the DB

```
#!/usr/bin/env php
// This file is part of Moodle - http://moodle.org/
//
// Moodle is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// Moodle is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with Moodle. If not, see <http://www.gnu.org/licenses/>.
```

VS Code Profile

```
        } catch (Exception $e) {
            flash("Error processing location change", "danger");
            error_log("Error processing location change: " . var_export($e, true));
        }
    } else {
        flash("Location is invalid, please try again.", "danger");
    }
} else {
    error_log("Location is invalid", "danger");
}

} <- #159-220 if ($can_update)
} <- #156-221 if (isset($_POST["location"]))
?>
```

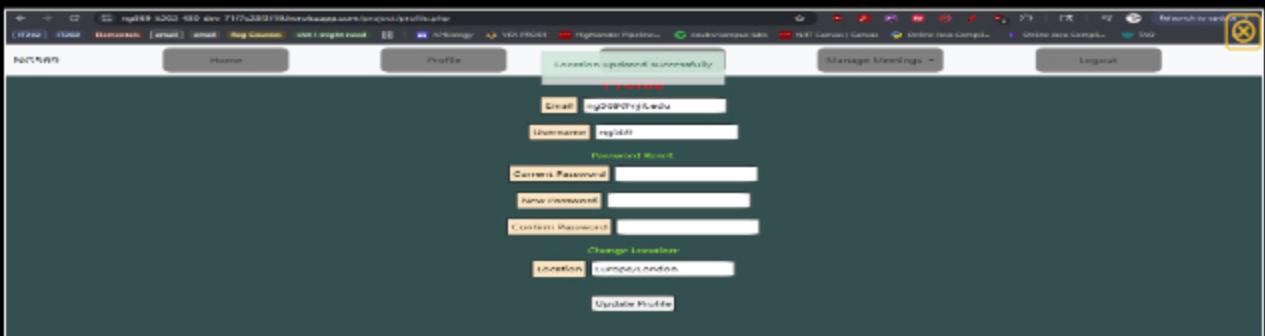
VS Code Profile 2

```
//location is placed into the API, where it will be checked to see if the API returns anything, if it does then it was a success and will go through an SQL call to be placed into the DB, if not then an error is returned
$location = ["location" => $location];
$endpoint = "https://worldtimeapi.org/api/v1/locations";
$location['location'] = $location['location'];
$location['tz_abb'] = $location['tz_abb'];
$location['tz_name'] = $location['tz_name'];
$location['tz_loc'] = $location['tz_loc'];
$location['gmt'] = $location['gmt'];
$location['tz_offset'] = $location['tz_offset'];
$location['registered_timezone'] = $location['registered_timezone'];

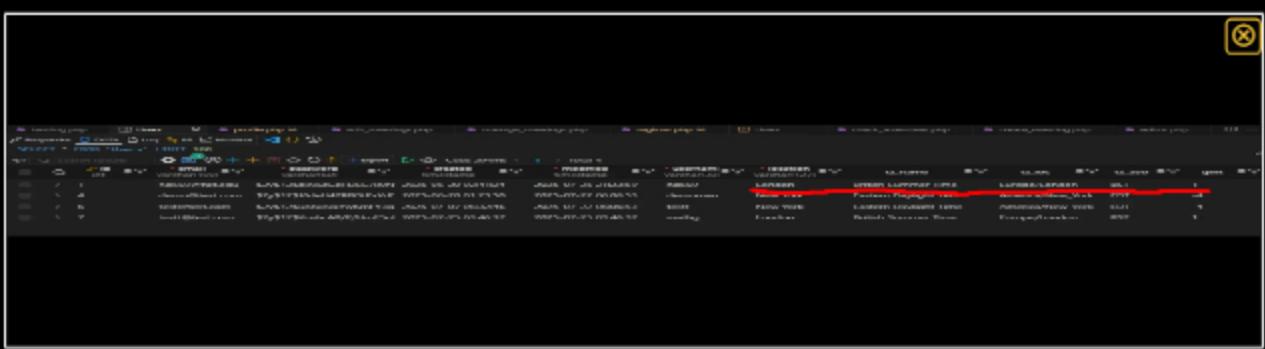
$location = json_encode($location);
$location = str_replace(' ', '+', $location);
```

```
"longitude":149.822281,  
"latitude":-34.8869572  
});  
  
error_log("Response: " . var_export($result, true));  
if ($result['status'] == 200 && !isset($result['response'])) {  
    $result = json_decode($result['response'], true);  
}  
else {  
    $result = [];  
    $msg = "Location was not recognized.";  
    $long = $lat;  
    $baseline = true;  
}  
return $result;
```

VS Code Register



Heroku Output



SQL Database



Saved: 7/25/2025 5:31:18 PM

⊕ Part 2:

Progress: 100%

Details:

- Include Heroku prod link to API fetch page
- Include pull request link for this feature

URL #1

<https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/profile.php>



URL

<https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/profile.php>



URL #2

<https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/register.php>



URL

<https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/register.php>



URL #3

<https://github.com/Nate-Gaw/ng569-IT202-4504>



URL

<https://github.com/Nate-Gaw/ng569-IT202-4504>



Part 3:

Progress: 100%

Details:

- Will you be using all the data or just a subset?
- Do you need to convert any data or extract pieces of a property's value?
- Briefly explain the transformation/extraction logic to get the API data to the shape your application intends to use

Your Response:

I only used a subset. I only grabbed what I needed from the API call, since everything else seemed unnecessary to grab and store. I do need to convert time based on other users' location and timezone, but that is done whenever needed, since there could be tens of different time zones and storing would feel inefficient. I needed the timezone, location, and abbreviation for the front-end sake, which allows the users to see where people are and the timezone for other people. I used mainly the GMT offset, since it was consistent with every location, in order to convert the timezone of the dates.



Saved: 7/25/2025 5:31:18 PM

Task #3 (0.50 pts.) - API Duplicates

Progress: 100%

Details:

- Consider how duplicate entries are handled
 - What happens if you get the same result from the API for an entity that exists but has been unmodified?
 - What happens if you get the same result from the API for an entity that exists but has been modified?

Your Response:

Since my api is used as a way to get data for each user. Duplicates are handled the same way as normal, since 2 people can be in the same location at one time. Same goes for if someone modifies their account in profile, the duplicate entry is still valid. Only way for it not to be valid is if there are 2 of the same accounts with the same location, but 2 accounts can not share the same email anyways so there wouldn't be any duplicate entry.



Saved: 7/25/2025 5:50:04 PM

Task #4 (0.50 pts.) - API Fetching

→ Task #4 (0.50 pts.) - AI II Etching

Progress: 100%

Details:

- Consider how your application will trigger this data fetch
 - (i.e., periodic via cron job, poor man's cron job, on app start/awake, etc)
 - (i.e., User or Admin triggered via a specific page/action)
 - (i.e., Lazy loaded during searches for stale/missing content)
 - What happens if you get the same result from the API for an entity that exists but has been modified?

Your Response:

The API is only triggered when the user updates their profile or registers an account. Then the location is inputted into the API, which then returns their location, time, and gmt offset, which I then extract. The API only requires the users location, so I have made it so the API is only called when the Users location is modified, which allows me to save on many api calls.



Saved: 7/25/2025 5:51:34 PM

Section #2: (1.5 pts.) Data Creation

Progress: 100%

☰ Task #1 (0.60 pts.) - Validations

Progress: 100%

Part 1:

Progress: 100%

Details:

- Show the html validations (code)
 - Show the javascript validations (code)
 - Show the php validations (code)



The screenshot shows a code editor with Java code. The code includes imports for `java.util.List`, `java.util.ArrayList`, and `java.util.stream.Collectors`. It defines a class named `Test` with a static method `main`. The method uses `Collectors.toList()` to convert a stream of integers to a list, which is then printed to the console.

```
import java.util.List;
import java.util.ArrayList;
import java.util.stream.Collectors;

public class Test {
    public static void main(String[] args) {
        List<Integer> list = Stream.of(1, 2, 3, 4, 5).collect(Collectors.toList());
        System.out.println(list);
    }
}
```

JS Validation

PHP Validation



Saved: 7/25/2025 6:56:48 PM

≡, Part 2:

Progress: 100%

Details:

- Briefly explain the html validations used
 - Briefly explain the javascript validation logic
 - Briefly explain the php validation logic

Your Response:

The html validations, JS Validations and php validations is similar. Its just checking to make sure that the inputs in the field are only letters and no special characters with some exception. Then the JS validation checks to see what needs to be updated and what is empty, meaning doesn't need to change.



Saved: 7/25/2025 6:56:48 PM

Task #2 (0.60 pts.) - Examples

Progress: 100%

Details:

- Show a successful creation

- Demonstrate various validation errors
- Ensure heroku url is visible

The screenshot shows a web page titled "Profile". The "Email" field contains "ng@ngtest.com" and has a red error message: "Email is required, you can't leave it blank". The "Password" field contains "password" and has a red error message: "Password is required". The "Confirm Password" field contains "password" and has a red error message: "Confirm Password". Below the fields are "Change location" and "Location" dropdown menus, both currently set to "London". At the bottom is a "Update Profile" button.

Successful creation Heroku

The screenshot shows a MySQL Workbench interface with a table named "Users". The table has columns: id, email, password, created, modified, username, location, tz_name, tz_loc, trabb, and gmt. There are 7 rows of data:

	ID	Email	Password	Created	Modified	Username	Location	TZ Name	TZ Loc	Trabb	Gmt
1	1	ng@ngtest.com	\$2y\$12\$pe7aH4CJmN	2025-06-20 07:41:24	2025-07-21 21:53:59	ng49	London	British Summer Time	Europe/London	EST	1
2	4	demo@ngtest.com	\$2y\$12\$6dU42B95PxW	2025-06-28 01:23:50	2025-07-22 06:06:53	demonen	New York	Eastern Daylight Time	America/New_York	EDT	-4
3	6	test@test.com	\$2y\$12\$uhwongYelmtY3q	2025-07-07 06:32:46	2025-07-22 05:05:53	test	New York	Eastern Daylight Time	America/New_York	EDT	-4
4	7	tech@test.com	\$2y\$12\$RoxVA87f4nEd5d	2025-07-25 03:46:37	2025-07-25 03:46:37	techy	London	British Summer Time	Europe/London	EST	1

Successful SQL DB creation

The screenshot shows a web page titled "Profile". The "Email" field contains "ng@ngtest.com" and has a red error message: "Email is required, you can't leave it blank". The "Password" field contains "password" and has a red error message: "Password is required". The "Confirm Password" field contains "password" and has a red error message: "Confirm Password". Below the fields are "Change location" and "Location" dropdown menus, both currently set to "London". At the bottom is a "Update Profile" button.

HTML Validation

The screenshot shows a web page titled "Profile". The "Email" field contains "ng@ngtest.com" and has a red error message: "Email is required, you can't leave it blank". The "Password" field contains "password" and has a red error message: "Password is required". The "Confirm Password" field contains "password" and has a red error message: "Confirm Password". Below the fields are "Change location" and "Location" dropdown menus, both currently set to "London". At the bottom is a "Update Profile" button.

JS Validation

The screenshot shows a web page titled "Profile". The "Email" field contains "ng@ngtest.com" and has a red error message: "Email is required, you can't leave it blank". The "Password" field contains "password" and has a red error message: "Password is required". The "Confirm Password" field contains "password" and has a red error message: "Confirm Password". Below the fields are "Change location" and "Location" dropdown menus, both currently set to "London". At the bottom is a "Update Profile" button.

Powered by:

Current Password: _____

New Password: _____

Confirm Password: _____

Change Location:

Latitude: _____ Longitude: _____

Update Profile

PHP Validation



Saved: 7/25/2025 6:56:34 PM

⌚ Task #3 (0.30 pts.) - Links

Progress: 100%

Details:

- Include the heroku prod link to this page
- Include pull request link for this feature

URL #1

<https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/profile.php>



URL

<https://ng569-it202-450-prod>



URL #2

<https://github.com/Nate-Gaw/ng569-IT202-450/pull/44>



URL

<https://github.com/Nate-Gaw>



Saved: 7/25/2025 6:57:37 PM

Section #3: (1.5 pts.) Data List Page (Many Items)

Progress: 100%

☰ Task #1 (0.60 pts.) - Code

Progress: 100%

💻 Part 1:

Progress: 100%

Details:

- Show the code that handles the filter/sort logic
- Show the code that generates the list/grid output
- Styling must be applied



```
SQL -> SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'EMPLOYEE';  
+-----+-----+-----+-----+-----+  
|   COLUMN_NAME   |   DATA_TYPE   |   IS_NULLABLE   |   COLUMN_TYPE   |   COLUMN_COMMENT   |  
+-----+-----+-----+-----+-----+  
|       ID       |       INT      |       YES      |       NOT NULL  |       ID of Employee|  
|       NAME     |       NVARCHAR |       YES      |       NOT NULL  |       Name of Employee|  
|       SALARY    |       DECIMAL   |       YES      |       NOT NULL  |       Salary of Employee|  
|       DEPARTMENT_ID |       INT      |       YES      |       NOT NULL  |       Department ID of Employee|  
+-----+-----+-----+-----+-----+
```

VS Code (Landing page)

```
SQL -> SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'EMPLOYEE';  
+-----+-----+-----+-----+-----+  
|   COLUMN_NAME   |   DATA_TYPE   |   IS_NULLABLE   |   COLUMN_TYPE   |   COLUMN_COMMENT   |  
+-----+-----+-----+-----+-----+  
|       ID       |       INT      |       YES      |       NOT NULL  |       ID of Employee|  
|       NAME     |       NVARCHAR |       YES      |       NOT NULL  |       Name of Employee|  
|       SALARY    |       DECIMAL   |       YES      |       NOT NULL  |       Salary of Employee|  
|       DEPARTMENT_ID |       INT      |       YES      |       NOT NULL  |       Department ID of Employee|  
+-----+-----+-----+-----+-----+
```

VS Code 1

```
SQL -> SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'EMPLOYEE';  
+-----+-----+-----+-----+-----+  
|   COLUMN_NAME   |   DATA_TYPE   |   IS_NULLABLE   |   COLUMN_TYPE   |   COLUMN_COMMENT   |  
+-----+-----+-----+-----+-----+  
|       ID       |       INT      |       YES      |       NOT NULL  |       ID of Employee|  
|       NAME     |       NVARCHAR |       YES      |       NOT NULL  |       Name of Employee|  
|       SALARY    |       DECIMAL   |       YES      |       NOT NULL  |       Salary of Employee|  
|       DEPARTMENT_ID |       INT      |       YES      |       NOT NULL  |       Department ID of Employee|  
+-----+-----+-----+-----+-----+
```

VS Code 2

```
SQL -> SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'EMPLOYEE';  
+-----+-----+-----+-----+-----+  
|   COLUMN_NAME   |   DATA_TYPE   |   IS_NULLABLE   |   COLUMN_TYPE   |   COLUMN_COMMENT   |  
+-----+-----+-----+-----+-----+  
|       ID       |       INT      |       YES      |       NOT NULL  |       ID of Employee|  
|       NAME     |       NVARCHAR |       YES      |       NOT NULL  |       Name of Employee|  
|       SALARY    |       DECIMAL   |       YES      |       NOT NULL  |       Salary of Employee|  
|       DEPARTMENT_ID |       INT      |       YES      |       NOT NULL  |       Department ID of Employee|  
+-----+-----+-----+-----+-----+
```

VS Code 3

```
SQL -> SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'EMPLOYEE';  
+-----+-----+-----+-----+-----+  
|   COLUMN_NAME   |   DATA_TYPE   |   IS_NULLABLE   |   COLUMN_TYPE   |   COLUMN_COMMENT   |  
+-----+-----+-----+-----+-----+  
|       ID       |       INT      |       YES      |       NOT NULL  |       ID of Employee|  
|       NAME     |       NVARCHAR |       YES      |       NOT NULL  |       Name of Employee|  
|       SALARY    |       DECIMAL   |       YES      |       NOT NULL  |       Salary of Employee|  
|       DEPARTMENT_ID |       INT      |       YES      |       NOT NULL  |       Department ID of Employee|  
+-----+-----+-----+-----+-----+
```



Saved: 7/25/2025 7:08:13 PM

≡, Part 2:

Progress: 100%

Details:

- Briefly explain how the filter/sort logic works and formulates the results
- Briefly explain how the list/grid output is generated
- Briefly note the styling choices

Your Response:

The pictures above show the landing page. Many of my other pages have similar layout and works, but they all serve a different purpose. The landing page was the most code-intensive for me. So the filtering works using JS. By using an event listener, I am able to grab anything placed into one of the search columns and display only information that contains the letters in the search bar. I was going to use PHP in order to sort, but I didn't like how PHP required a reload of the page almost every time, and found JS was able to do it automatically. But in this case, if the row contains what the user is filtering, then the row will be queued and outputted onto the users screen. I decided to create 5 different search bars, as it will allow the user to have an easier time filtering through the meetings, rather than just one or two and having to struggle.

As for the below, there is a separate page I will show where the user can go and edit their own meetings, as I didn't want the landing page to be so cluttered. And the links are embedded into the row. So the edit and delete row is a button, but to see details is just a simple click on the row. This is applied to all my meeting tables.



Saved: 7/25/2025 7:08:13 PM

▣ Task #2 (0.60 pts.) - Examples

Progress: 100%

Details:

- Show a few variations of filter/sort applied (including no records)
- Each list item should be a summary
- Each list item should have the following links
 - A link to a single view of the specific entity (i.e., a details page)
 - A link to delete this entity (this may be an admin-only functionality, but it should be present for the respective role)
 - A link to edit this entity (this may be an admin-only functionality, but it should be present for the respective role)
- Ensure heroku url is visible

Meetings				
ID	Name	Creator	Message	Date & Time
1	Meeting 1	John Doe		2023-07-01 10:00:00 UTC+00:00
2	Meeting 2	John Doe		2023-07-01 10:00:00 UTC+00:00
3	Meeting 3	John Doe		2023-07-01 10:00:00 UTC+00:00
4	Meeting 4	John Doe	Meeting 4 description	2023-07-01 10:00:00 UTC+00:00
5	Meeting 5	John Doe	Meeting 5 description	2023-07-01 10:00:00 UTC+00:00
6	Meeting 6	John Doe	Meeting 6 description	2023-07-01 10:00:00 UTC+00:00
7	Meeting 7	John Doe	Meeting 7 description	2023-07-01 10:00:00 UTC+00:00
8	Meeting 8	John Doe	Meeting 8 description	2023-07-01 10:00:00 UTC+00:00
9	Meeting 9	John Doe	Meeting 9 description	2023-07-01 10:00:00 UTC+00:00
10	Meeting 10	John Doe	Meeting 10 description	2023-07-01 10:00:00 UTC+00:00

Heroku Empty Filter

Meetings				
ID	Name	Creator	Message	Date & Time
1	Meeting 1	John Doe	Meeting 1 description	2023-07-01 10:00:00 UTC+00:00

Heroku Creator Filter

Meetings				
ID	Name	Creator	Message	Date & Time
1	Meeting 1	John Doe	Meeting 1 description	2023-07-01 10:00:00 UTC+00:00
2	Meeting 2	John Doe	Meeting 2 description	2023-07-01 10:00:00 UTC+00:00
3	Meeting 3	John Doe	Meeting 3 description	2023-07-01 10:00:00 UTC+00:00
4	Meeting 4	John Doe	Meeting 4 description	2023-07-01 10:00:00 UTC+00:00
5	Meeting 5	John Doe	Meeting 5 description	2023-07-01 10:00:00 UTC+00:00
6	Meeting 6	John Doe	Meeting 6 description	2023-07-01 10:00:00 UTC+00:00
7	Meeting 7	John Doe	Meeting 7 description	2023-07-01 10:00:00 UTC+00:00
8	Meeting 8	John Doe	Meeting 8 description	2023-07-01 10:00:00 UTC+00:00
9	Meeting 9	John Doe	Meeting 9 description	2023-07-01 10:00:00 UTC+00:00
10	Meeting 10	John Doe	Meeting 10 description	2023-07-01 10:00:00 UTC+00:00

Heroku Date and Time Filter

Meetings				
ID	Name	Creator	Message	Date & Time
1	Meeting 1	John Doe	Meeting 1 description	2023-07-01 10:00:00 UTC+00:00
2	Meeting 2	John Doe	Meeting 2 description	2023-07-01 10:00:00 UTC+00:00
3	Meeting 3	John Doe	Meeting 3 description	2023-07-01 10:00:00 UTC+00:00
4	Meeting 4	John Doe	Meeting 4 description	2023-07-01 10:00:00 UTC+00:00
5	Meeting 5	John Doe	Meeting 5 description	2023-07-01 10:00:00 UTC+00:00
6	Meeting 6	John Doe	Meeting 6 description	2023-07-01 10:00:00 UTC+00:00
7	Meeting 7	John Doe	Meeting 7 description	2023-07-01 10:00:00 UTC+00:00
8	Meeting 8	John Doe	Meeting 8 description	2023-07-01 10:00:00 UTC+00:00
9	Meeting 9	John Doe	Meeting 9 description	2023-07-01 10:00:00 UTC+00:00
10	Meeting 10	John Doe	Meeting 10 description	2023-07-01 10:00:00 UTC+00:00

Heroku Edit Meeting Page (Filter works the same)

Meetings				
ID	Name	Creator	Message	Date & Time
1	Meeting 1	John Doe	Meeting 1 description	2023-07-01 10:00:00 UTC+00:00

Heroku No results

The screenshot shows a web browser window with a dark theme. At the top, there's a navigation bar with various icons and a search bar. Below it, a header says "Home Page" and "Meetings". A sidebar on the left lists "Your Roles" with "Admin" and "Meeting" selected. The main content area has a heading "Search:" followed by a table with four columns: "Meet Name", "Status", "Message", and "Date & Time". There are three rows of data:

Meet Name	Status	Message	Date & Time
Meeting 1	Upcoming	Meeting 1 is coming up.	2025-07-25 14:15:00
Meeting 2	Upcoming	Meeting 2 is coming up.	2025-07-25 14:15:00

Heroku Index Filter

This screenshot is similar to the first one but shows a different set of data in the table. The table has the same structure:

Meet Name	Status	Message	Date & Time
Meeting 1	Upcoming	Meeting 1 is coming up.	2025-07-25 14:15:00
Meeting 2	Upcoming	Meeting 2 is coming up.	2025-07-25 14:15:00
Meeting 3	Upcoming	Meeting 3 is coming up.	2025-07-25 14:15:00
Meeting 4	Upcoming	Meeting 4 is coming up.	2025-07-25 14:15:00

Heroku Message Filter

This screenshot shows a filtered view of the meeting data. The table now only contains the first two rows from the previous ones:

Meet Name	Status	Message	Date & Time
Meeting 1	Upcoming	Meeting 1 is coming up.	2025-07-25 14:15:00
Meeting 2	Upcoming	Meeting 2 is coming up.	2025-07-25 14:15:00

Heroku Row Filter

A blue status bar at the bottom of the screen displays a save icon and the text "Saved: 7/25/2025 7:14:09 PM".

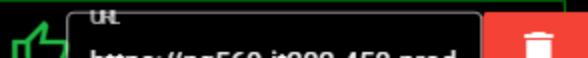
☞ Task #3 (0.30 pts.) - Links

Progress: 100%

Details:

- Include the heroku prod link to this page
- Include pull request link for this feature

URL #1



<https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/landing.php>

URL #2

https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/edit_meetings.php



1

https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/edit_meetings.php



Saved: 7/25/2025 7:14:48 PM

Section #4: (1.5 pts.) View Details Page (Single Item)

Progress: 100%

≡ Task #1 (0.60 pts.) - Code

Progress: 100%

Part 1:

Progress: 100%

Details:

- Show the code that handles loading of the record (and how the id retrieved for usage)
 - Include the code that handles invalid/missing ids
- Show the code that generates the entity output
- Styling must be applied

A screenshot of the Visual Studio Code interface. The main editor window shows several lines of code, likely PHP or similar, with syntax highlighting. A status bar at the bottom indicates "VS Code Loading in record user inputs". The interface includes standard VS Code elements like the file menu, search bar, and sidebar.

VS Code Loading in record user inputs

A screenshot of the Visual Studio Code interface, similar to the previous one. The main editor window shows code, and the status bar at the bottom indicates "HTML output". The interface includes standard VS Code elements like the file menu, search bar, and sidebar.

HTML output



HTML output 2



Saved: 7/25/2025 7:22:05 PM

☞ Part 2:

Progress: 100%

Details:

- Briefly explain how the loading logic works
- Briefly explain how the entity output is generated
- Briefly note the styling choices

Your Response:

The loading logic waits for the user input. Then if the user leaves it empty it will error, but if they put in a valid number. The code checks if the user has access to the meeting or is an admin, preventing others from scanning random meeting details. Then the index is loaded into the sql and all relevant information is grabbed from attendees to the meeting time and date. Then simple html outputs all the data in 2 tables, one for general meeting information and one for attendees invited.

For below, editing and deleting is only on you manage meeting page so that attendees cannot delete other people's meeting unless they are the host or admin. Same goes for deleting. I didn't want anyone to be able to modify or change around the meeting information unless they were admins or the owner of the meeting



Saved: 7/25/2025 7:22:05 PM

▣ Task #2 (0.60 pts.) - Examples

Progress: 100%

Details:

- Show a few examples of different entities
- Each entity item should have more details than the summary view (if possible)
- Each entity should have the following links
 - A link to edit the single entity (this may be an admin-only thing, but it should be present for the respective role)
 - A link to delete the single entity (this may be an admin-only thing, but it

should be present for the respective role)

Check Meeting Details

Meeting ID	Creator	Message	Date & Time	Original Date & Time - GMT
6	nategaw	Testing creating a meeting from a different account. Failed the meeting	2023-01-29 23:05:05	2023-01-29 11:05:05

Attendees:

Username	Email	Location
nategaw	nategaw@outlook.com	America/New_York
ng569	ng569@outlook.com	America/New_York
nategaw	nategaw@outlook.com	America/New_York

Heroku output for lookup

Check Meeting Details

Meeting ID	Creator	Message	Date & Time	Original Date & Time - GMT
19	nategaw	Test has everyone in it	2023-01-26 16:12:10	2023-01-26 17:12:09

Attendees:

Username	Email	Location
nategaw	nategaw@outlook.com	America/New_York
ng569	ng569@outlook.com	America/New_York
nategaw	nategaw@outlook.com	America/New_York
ng569	ng569@outlook.com	America/New_York

Heroku output from redirect

Saved: 7/25/2025 7:23:32 PM

☞ Task #3 (0.30 pts.) - Links

Progress: 100%

Details:

- Include the heroku prod link to this page
- Include pull request link for this feature

URL #1
https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/check_attendees.php

URL #2
<https://github.com/Nate-Gaw/ng569-IT202-450/pull/44>

Saved: 7/25/2025 7:24:00 PM

Section #5: (1.5 pts.) Edit Page

Progress: 100%

☰ Task #1 (0.60 pts.) - Code

Progress: 100%

☒ Part 1:

Progress: 100%

Details:

- Show the code that handles loading of the record (and how the id retrieved for usage)
 - Include the code that handles invalid/missing ids
- Show the code that generates the form with the correct data types and populates it
- Show the html, javascript, and php validation (should be similar to create)
- Styling must be applied

```
function loadRecord($id) {
    $query = "SELECT * FROM users WHERE id = ?";
```

This screenshot shows a code editor window with PHP code. The code defines a function named 'loadRecord' that takes an argument '\$id'. Inside the function, there is a SQL query to select all columns from the 'users' table where the 'id' matches the value of '\$id'. The code uses question marks as placeholders for the value of '\$id'.

Edit Page Code

```
<?php echo $form->input('username', array('label' => 'Username', 'value' => $user['username'])); ?>
```

This screenshot shows a code editor window with PHP code. It includes a line of code that outputs an input field for a 'username' field. The code uses the 'input' method of a variable named '\$form'. The 'label' attribute is set to 'Username' and the 'value' attribute is set to the value of '\$user['username']'.

Edit Page Code

```
if ($form->isSubmitted() && $form->isValid()) {
    $user = $form->getValues();
    $user['password'] = password_hash($user['password'], PASSWORD_DEFAULT);
    $user['id'] = null;
    $user['created_at'] = null;
    $user['updated_at'] = null;
}
```

This screenshot shows a code editor window with PHP code. It contains an if statement that checks if the form has been submitted and is valid. If so, it retrieves the values from the form, hashes the password using 'password_hash' with the 'PASSWORD_DEFAULT' option, and sets the 'id' and 'created_at' fields to null. It also sets the 'updated_at' field to null.

Edit Page Code

```
function updateRecord($user) {
    $query = "UPDATE users SET ... WHERE id = ?";
```

This screenshot shows a code editor window with PHP code. It includes a line of code that starts a query to update the 'users' table. The code uses question marks as placeholders for the value of '\$user'. The rest of the code is partially visible at the bottom of the screen.

```
git clone https://github.com/.../Meeting-Platform.git
```

Edit Page Code

```
git clone https://github.com/.../Meeting-Platform.git
```

Edit Page Code



=, Part 2:

Progress: 100%

Details:

- Briefly explain how the loading logic works
- Briefly explain how the form is generated and submission is handled
- Briefly note the styling choices
- Note what fields are editable and why the choice was made for them (it's likely not all fields are editable, especially `id`, `created`, `modified`, etc)

Your Response:

So this page is the most load intensive. First, the page is redirected with the meeting ID in the GET. Then the page has to rip all the information from the meeting ID plus the attendees, to have the checkboxes for the attendees filled in. Then, a normal call to the Users db is made in order to create the list of users of the platform. Then the page works similar to the create page, where the user will change and fill out anything that wasn't before. Then once the form is filled, there are a couple of calls to change the meeting information, as well as the attendee list, which takes up about 3 calls, due to the need to insert and delete attendees, as well as change meeting information in several DBs.



▣ Task #2 (0.60 pts.) - Examples

Progress: 100%

Details:

- Show a before and after of updating an entity (clearly captioned)
- Successful update should have an appropriate message shown
- The updated data should be shown in the form
- Any errors should have user-friendly messages shown
- Show some examples of validation errors (html, javascript, php)

The screenshot shows a web-based application titled "Manage Your Meetings". The main area displays a table of meetings with columns for "Meeting ID", "Meeting Name", "Description", "Start Date/Time", "End Date/Time", "Location", and "Status". One meeting, identified by ID 10, is highlighted with a yellow border. The status for this meeting is "Upcoming". Below the table, there is a "Edit" button next to the highlighted row.

Before change (changing id 10)

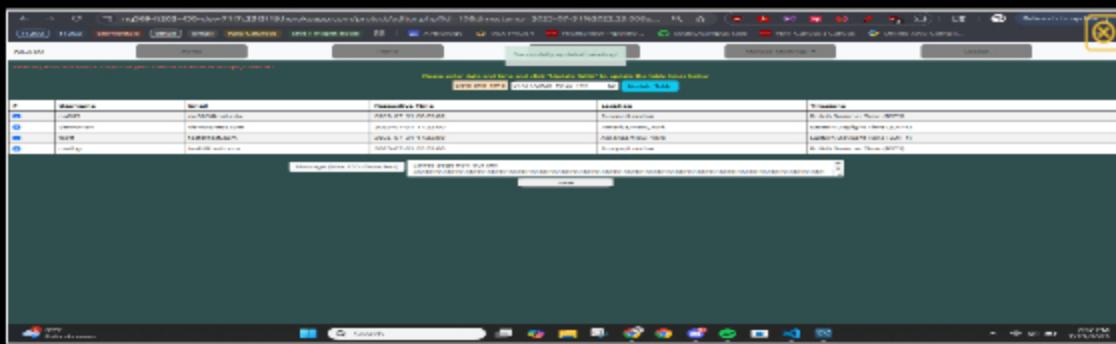
This screenshot shows the same web application after the meeting with ID 10 has been updated. The status of the meeting is now "Completed". The "Edit" button is still present below the table.

Initial edit page

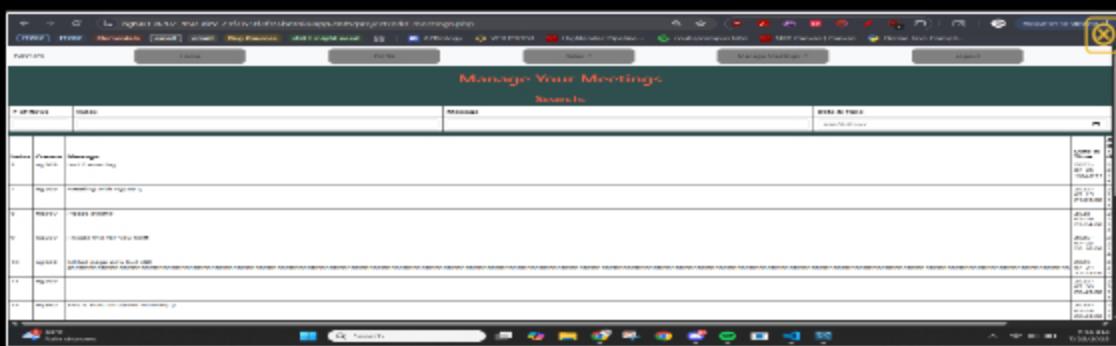
The screenshot shows a web-based application with a table of users. One user, identified by ID 10, is highlighted with a yellow border. The user's name is "John Doe" and their email is "john.doe@example.com". Below the table, there is a "Edit" button next to the highlighted row.

HTML validation

The screenshot shows a user profile edit form. There are several input fields: "First Name" (John), "Last Name" (Doe), "Permanent Address" (123 Main St, Anytown, USA), "Current Password" (password), "New Password" (newpassword), and "Confirm Password" (newpassword). Below these fields are two error messages: "The current password is required" and "The new password does not match the confirmation". At the bottom of the form is a "Update Profile" button.



Success message



Finished Edit



Saved: 7/25/2025 7:40:01 PM

⌚ Task #3 (0.30 pts.) - Links

Progress: 100%

Details:

- Include the heroku prod link to this page
- Include pull request link for this feature

URL #1

https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/edit_meetings.php



<https://ng569-it202-450-prod>



URL #2

<https://github.com/Nate-Gaw/ng569-IT202-450/pull/44>



<https://github.com/Nate-Gaw>



Saved: 7/25/2025 7:40:40 PM

Section #6: (1.5 pts.) Delete Logic

Progress: 100%

≡ Task #1 (0.60 pts.) - Code

Progress: 100%

Part 1:

Progress: 100%

Details:

- Show the code that handles deletion of the record (and how the id retrieved for usage)
 - Include the code that handles invalid/missing ids
- Handle any necessary roles/permissions (i.e., it's not likely that anyone can delete any entity they choose)
- Show the redirect logic that goes back to the previous page (where the route came from), if it was a list page, it should maintain the filter/sort criteria

```
//handle delete
//ng569 7/25/2025
//Grab post request as well as delete id sent, and delete the row. DB is cascade so every attendee is removed as well
if ($_SERVER["REQUEST_METHOD"] === "POST" && isset($_POST["delete_id"])) {
    $id = $_POST["delete_id"];
    $stmt2 = $db->prepare("DELETE FROM Meetings WHERE id = :id");
    try {
        $stmt2->execute([":id" => $id]);
        header("Location: " . $_SERVER['PHP_SELF'] . "?success=true"); //reload page
    } catch (PDOException $e) {
        flash("There was an error deleting meetings, please contact an admin for support Code 6", "danger");
        error_log("Error deleting meetings" . var_export($e->errorInfo, true));
    }
} <- #30-40 if ($_SERVER["REQUEST_METHOD"] === "POST" && isset($_POST["de...")
```

Delete Button PHP

```
<?php foreach ($meetings as $Meeting) { ?>
    <?php echo "check_attendees.php/index.php echo Meeting('id'); ?>
        <?php echo $Meeting['id']; ?></td>
        <?php echo $Meeting['host'; ?></td>
        <?php echo $Meeting['message'; ?></td>
        <?php
            <?php
                echo convertTimezone($Meeting['MeetingDate'], $Meeting['set'], get_user.get());
            </?php
        </?td>
        <?php
            echo $Meeting['MeetingDate'] . '-' . $Meeting['set'];
        </?td>
        <?php
            $action = "use atm warning" . $Meeting['MeetingDate'] . '-' . $Meeting['set'] . "-warning";
            $cancel = "use atm warning" . $Meeting['MeetingDate'] . '-' . $Meeting['set'] . "-cancel";
            $submit = "use atm warning" . $Meeting['MeetingDate'] . '-' . $Meeting['set'] . "-submit";
            $button = "use atm warning" . $Meeting['MeetingDate'] . '-' . $Meeting['set'] . "-button";
        </?td>
        <?td>
            <form method="POST">
                <input type="hidden" name="delete_id" value="<?php echo $Meeting['id'; ?>">
                <button id="Delete" type="submit" class="btn btn-danger">Delete</button>
            </form>
        </?td>
    </?tr>
```

Delete Button HTML

 Saved: 7/25/2025 7:44:51 PM

≡ Part 2:

Progress: 100%

Details:

- Briefly explain how the deletion logic works
- Is it a soft or hard delete?
- Note any permissions/rules applied.
- Briefly explain how the redirect maintains the previous route and any filter/sort data

Your Response:

I just made the delete part a button. I see no reason to make it any more complicated, as I have an edit page too. It is a hard delte, which will delete the meeting from the meeting db as well as deleting all attendees since the db is cascade, which will delete anything associated with it.



Saved: 7/25/2025 7:44:51 PM

Task #2 (0.60 pts.) - Examples

Progress: 100%

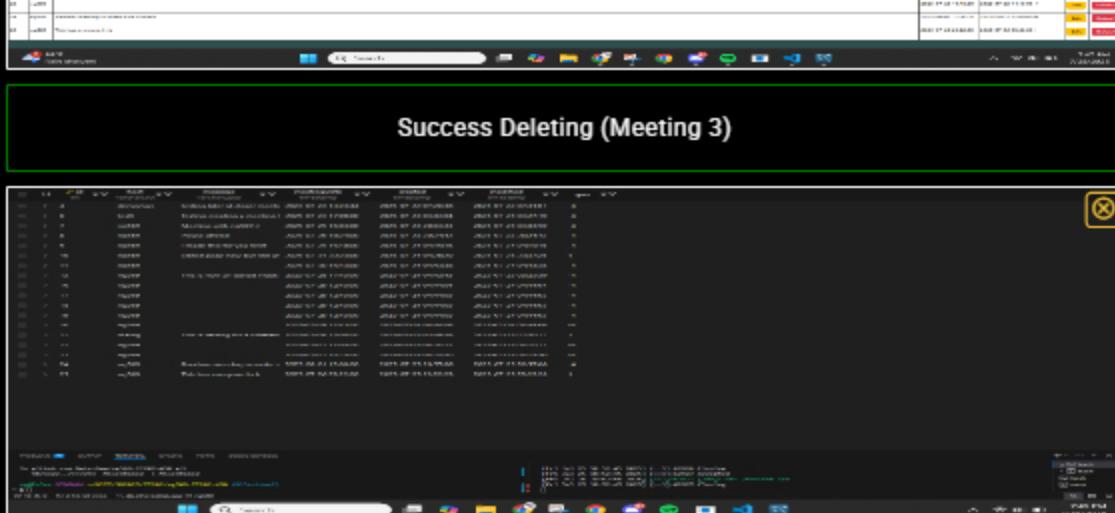
Details:

- Show a successful delete message
- Show a before and after database view of the record being deleted (clearly captioned)
- Successful update should have an appropriate message shown
- The updated data should be shown in the form
- Any errors should have user-friendly messages shown

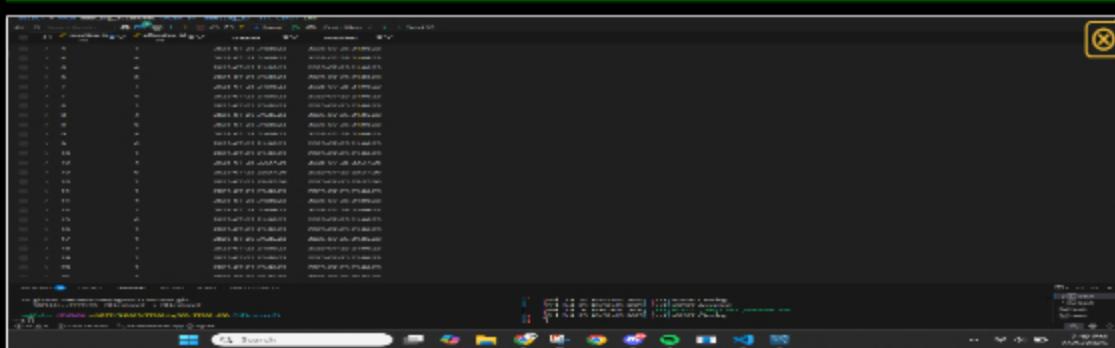
ID	Title	Description	Location	Date	Start Time	End Time	Attendees	Status
1	Meeting 1	Meeting 1 Description	Office 1	2025-07-25	10:00:00	11:00:00	John Doe, Jane Smith	Pending
2	Meeting 2	Meeting 2 Description	Office 2	2025-07-26	14:00:00	15:00:00	John Doe, Jane Smith	Pending
3	Meeting 3	Meeting 3 Description	Office 3	2025-07-27	09:00:00	10:00:00	John Doe, Jane Smith	Pending
4	Meeting 4	Meeting 4 Description	Office 4	2025-07-28	13:00:00	14:00:00	John Doe, Jane Smith	Pending
5	Meeting 5	Meeting 5 Description	Office 5	2025-07-29	08:00:00	09:00:00	John Doe, Jane Smith	Pending
6	Meeting 6	Meeting 6 Description	Office 6	2025-07-30	12:00:00	13:00:00	John Doe, Jane Smith	Pending
7	Meeting 7	Meeting 7 Description	Office 7	2025-07-31	15:00:00	16:00:00	John Doe, Jane Smith	Pending
8	Meeting 8	Meeting 8 Description	Office 8	2025-08-01	09:00:00	10:00:00	John Doe, Jane Smith	Pending
9	Meeting 9	Meeting 9 Description	Office 9	2025-08-02	14:00:00	15:00:00	John Doe, Jane Smith	Pending
10	Meeting 10	Meeting 10 Description	Office 10	2025-08-03	08:00:00	09:00:00	John Doe, Jane Smith	Pending
11	Meeting 11	Meeting 11 Description	Office 11	2025-08-04	13:00:00	14:00:00	John Doe, Jane Smith	Pending
12	Meeting 12	Meeting 12 Description	Office 12	2025-08-05	09:00:00	10:00:00	John Doe, Jane Smith	Pending
13	Meeting 13	Meeting 13 Description	Office 13	2025-08-06	14:00:00	15:00:00	John Doe, Jane Smith	Pending
14	Meeting 14	Meeting 14 Description	Office 14	2025-08-07	08:00:00	09:00:00	John Doe, Jane Smith	Pending
15	Meeting 15	Meeting 15 Description	Office 15	2025-08-08	13:00:00	14:00:00	John Doe, Jane Smith	Pending
16	Meeting 16	Meeting 16 Description	Office 16	2025-08-09	09:00:00	10:00:00	John Doe, Jane Smith	Pending
17	Meeting 17	Meeting 17 Description	Office 17	2025-08-10	14:00:00	15:00:00	John Doe, Jane Smith	Pending
18	Meeting 18	Meeting 18 Description	Office 18	2025-08-11	08:00:00	09:00:00	John Doe, Jane Smith	Pending
19	Meeting 19	Meeting 19 Description	Office 19	2025-08-12	13:00:00	14:00:00	John Doe, Jane Smith	Pending
20	Meeting 20	Meeting 20 Description	Office 20	2025-08-13	09:00:00	10:00:00	John Doe, Jane Smith	Pending
21	Meeting 21	Meeting 21 Description	Office 21	2025-08-14	14:00:00	15:00:00	John Doe, Jane Smith	Pending
22	Meeting 22	Meeting 22 Description	Office 22	2025-08-15	08:00:00	09:00:00	John Doe, Jane Smith	Pending
23	Meeting 23	Meeting 23 Description	Office 23	2025-08-16	13:00:00	14:00:00	John Doe, Jane Smith	Pending
24	Meeting 24	Meeting 24 Description	Office 24	2025-08-17	09:00:00	10:00:00	John Doe, Jane Smith	Pending
25	Meeting 25	Meeting 25 Description	Office 25	2025-08-18	14:00:00	15:00:00	John Doe, Jane Smith	Pending
26	Meeting 26	Meeting 26 Description	Office 26	2025-08-19	08:00:00	09:00:00	John Doe, Jane Smith	Pending
27	Meeting 27	Meeting 27 Description	Office 27	2025-08-20	13:00:00	14:00:00	John Doe, Jane Smith	Pending
28	Meeting 28	Meeting 28 Description	Office 28	2025-08-21	09:00:00	10:00:00	John Doe, Jane Smith	Pending
29	Meeting 29	Meeting 29 Description	Office 29	2025-08-22	14:00:00	15:00:00	John Doe, Jane Smith	Pending
30	Meeting 30	Meeting 30 Description	Office 30	2025-08-23	08:00:00	09:00:00	John Doe, Jane Smith	Pending
31	Meeting 31	Meeting 31 Description	Office 31	2025-08-24	13:00:00	14:00:00	John Doe, Jane Smith	Pending
32	Meeting 32	Meeting 32 Description	Office 32	2025-08-25	09:00:00	10:00:00	John Doe, Jane Smith	Pending
33	Meeting 33	Meeting 33 Description	Office 33	2025-08-26	14:00:00	15:00:00	John Doe, Jane Smith	Pending
34	Meeting 34	Meeting 34 Description	Office 34	2025-08-27	08:00:00	09:00:00	John Doe, Jane Smith	Pending
35	Meeting 35	Meeting 35 Description	Office 35	2025-08-28	13:00:00	14:00:00	John Doe, Jane Smith	Pending
36	Meeting 36	Meeting 36 Description	Office 36	2025-08-29	09:00:00	10:00:00	John Doe, Jane Smith	Pending
37	Meeting 37	Meeting 37 Description	Office 37	2025-08-30	14:00:00	15:00:00	John Doe, Jane Smith	Pending
38	Meeting 38	Meeting 38 Description	Office 38	2025-08-31	08:00:00	09:00:00	John Doe, Jane Smith	Pending
39	Meeting 39	Meeting 39 Description	Office 39	2025-09-01	13:00:00	14:00:00	John Doe, Jane Smith	Pending
40	Meeting 40	Meeting 40 Description	Office 40	2025-09-02	09:00:00	10:00:00	John Doe, Jane Smith	Pending
41	Meeting 41	Meeting 41 Description	Office 41	2025-09-03	14:00:00	15:00:00	John Doe, Jane Smith	Pending
42	Meeting 42	Meeting 42 Description	Office 42	2025-09-04	08:00:00	09:00:00	John Doe, Jane Smith	Pending
43	Meeting 43	Meeting 43 Description	Office 43	2025-09-05	13:00:00	14:00:00	John Doe, Jane Smith	Pending
44	Meeting 44	Meeting 44 Description	Office 44	2025-09-06	09:00:00	10:00:00	John Doe, Jane Smith	Pending
45	Meeting 45	Meeting 45 Description	Office 45	2025-09-07	14:00:00	15:00:00	John Doe, Jane Smith	Pending
46	Meeting 46	Meeting 46 Description	Office 46	2025-09-08	08:00:00	09:00:00	John Doe, Jane Smith	Pending
47	Meeting 47	Meeting 47 Description	Office 47	2025-09-09	13:00:00	14:00:00	John Doe, Jane Smith	Pending
48	Meeting 48	Meeting 48 Description	Office 48	2025-09-10	09:00:00	10:00:00	John Doe, Jane Smith	Pending
49	Meeting 49	Meeting 49 Description	Office 49	2025-09-11	14:00:00	15:00:00	John Doe, Jane Smith	Pending
50	Meeting 50	Meeting 50 Description	Office 50	2025-09-12	08:00:00	09:00:00	John Doe, Jane Smith	Pending
51	Meeting 51	Meeting 51 Description	Office 51	2025-09-13	13:00:00	14:00:00	John Doe, Jane Smith	Pending
52	Meeting 52	Meeting 52 Description	Office 52	2025-09-14	09:00:00	10:00:00	John Doe, Jane Smith	Pending
53	Meeting 53	Meeting 53 Description	Office 53	2025-09-15	14:00:00	15:00:00	John Doe, Jane Smith	Pending
54	Meeting 54	Meeting 54 Description	Office 54	2025-09-16	08:00:00	09:00:00	John Doe, Jane Smith	Pending
55	Meeting 55	Meeting 55 Description	Office 55	2025-09-17	13:00:00	14:00:00	John Doe, Jane Smith	Pending
56	Meeting 56	Meeting 56 Description	Office 56	2025-09-18	09:00:00	10:00:00	John Doe, Jane Smith	Pending
57	Meeting 57	Meeting 57 Description	Office 57	2025-09-19	14:00:00	15:00:00	John Doe, Jane Smith	Pending
58	Meeting 58	Meeting 58 Description	Office 58	2025-09-20	08:00:00	09:00:00	John Doe, Jane Smith	Pending
59	Meeting 59	Meeting 59 Description	Office 59	2025-09-21	13:00:00	14:00:00	John Doe, Jane Smith	Pending
60	Meeting 60	Meeting 60 Description	Office 60	2025-09-22	09:00:00	10:00:00	John Doe, Jane Smith	Pending
61	Meeting 61	Meeting 61 Description	Office 61	2025-09-23	14:00:00	15:00:00	John Doe, Jane Smith	Pending
62	Meeting 62	Meeting 62 Description	Office 62	2025-09-24	08:00:00	09:00:00	John Doe, Jane Smith	Pending
63	Meeting 63	Meeting 63 Description	Office 63	2025-09-25	13:00:00	14:00:00	John Doe, Jane Smith	Pending
64	Meeting 64	Meeting 64 Description	Office 64	2025-09-26	09:00:00	10:00:00	John Doe, Jane Smith	Pending
65	Meeting 65	Meeting 65 Description	Office 65	2025-09-27	14:00:00	15:00:00	John Doe, Jane Smith	Pending
66	Meeting 66	Meeting 66 Description	Office 66	2025-09-28	08:00:00	09:00:00	John Doe, Jane Smith	Pending
67	Meeting 67	Meeting 67 Description	Office 67	2025-09-29	13:00:00	14:00:00	John Doe, Jane Smith	Pending
68	Meeting 68	Meeting 68 Description	Office 68	2025-09-30	09:00:00	10:00:00	John Doe, Jane Smith	Pending
69	Meeting 69	Meeting 69 Description	Office 69	2025-10-01	14:00:00	15:00:00	John Doe, Jane Smith	Pending
70	Meeting 70	Meeting 70 Description	Office 70	2025-10-02	08:00:00	09:00:00	John Doe, Jane Smith	Pending
71	Meeting 71	Meeting 71 Description	Office 71	2025-10-03	13:00:00	14:00:00	John Doe, Jane Smith	Pending
72	Meeting 72	Meeting 72 Description	Office 72	2025-10-04	09:00:00	10:00:00	John Doe, Jane Smith	Pending
73	Meeting 73	Meeting 73 Description	Office 73	2025-10-05	14:00:00	15:00:00	John Doe, Jane Smith	Pending
74	Meeting 74	Meeting 74 Description	Office 74	2025-10-06	08:00:00	09:00:00	John Doe, Jane Smith	Pending
75	Meeting 75	Meeting 75 Description	Office 75	2025-10-07	13:00:00	14:00:00	John Doe, Jane Smith	Pending
76	Meeting 76	Meeting 76 Description	Office 76	2025-10-08	09:00:00	10:00:00	John Doe, Jane Smith	Pending
77	Meeting 77	Meeting 77 Description	Office 77	2025-10-09	14:00:00	15:00:00	John Doe, Jane Smith	Pending
78	Meeting 78	Meeting 78 Description	Office 78	2025-10-10	08:00:00	09:00:00	John Doe, Jane Smith	Pending
79	Meeting 79	Meeting 79 Description	Office 79	2025-10-11	13:00:00	14:00:00	John Doe, Jane Smith	Pending
80	Meeting 80	Meeting 80 Description	Office 80	2025-10-12	09:00:00	10:00:00	John Doe, Jane Smith	Pending
81	Meeting 81	Meeting 81 Description	Office 81	2025-10-13	14:00:00	15:00:00	John Doe, Jane Smith	Pending
82	Meeting 82	Meeting 82 Description	Office 82	2025-10-14	08:00:00	09:00:00	John Doe, Jane Smith	Pending
83	Meeting 83	Meeting 83 Description	Office 83	2025-10-15	13:00:00	14:00:00	John Doe, Jane Smith	Pending
84	Meeting 84	Meeting 84 Description	Office 84	2025-10-16	09:00:00	10:00:00	John Doe, Jane Smith	Pending
85	Meeting 85	Meeting 85 Description	Office 85	2025-10-17	14:00:00	15:00:00	John Doe, Jane Smith	Pending
86	Meeting 86	Meeting 86 Description	Office 86	2025-10-18	08:00:00	09:00:00	John Doe, Jane Smith	Pending
87	Meeting 87	Meeting 87 Description	Office 87	2025-10-19	13:00:00	14:00:00	John Doe, Jane Smith	Pending
88	Meeting 88	Meeting 88 Description	Office 88	2025-10-20	09:00:00	10:00:00	John Doe, Jane Smith	Pending
89	Meeting 89	Meeting 89 Description	Office 89	2025-10-21	14:00:00	15:00:00	John Doe, Jane Smith	Pending
90	Meeting 90	Meeting 90 Description	Office 90	2025-10-22	08:00:00	09:00:00	John Doe, Jane Smith	Pending
91	Meeting 91	Meeting 91 Description	Office 91	2025-10-23	13:00:00	14:00:00	John Doe, Jane Smith	Pending
92	Meeting 92	Meeting 92 Description	Office 92	2025-10-24	09:00:00	10:00:00	John Doe, Jane Smith	Pending
93	Meeting 93	Meeting 93 Description	Office 93	2025-10-25	14:00:00	15:00:00	John Doe, Jane Smith	Pending
94	Meeting 94	Meeting 94 Description	Office 94	2025-10-26	08:00:00	09:00:00	John Doe, Jane Smith	Pending
95	Meeting 95	Meeting 95 Description	Office 95	2025-10-27	13:00:00	14:00:00	John Doe, Jane Smith	Pending
96	Meeting 96	Meeting 96 Description	Office 96	2025-10-28	09:00:00	10:00:00	John Doe, Jane Smith	Pending
97	Meeting 97	Meeting 97 Description	Office 97	2025-10-29	14:00:00	15:00:00	John Doe, Jane Smith	Pending
98	Meeting 98	Meeting 98 Description	Office 98	2025-10-30	08:00:00	09:00:00	John Doe, Jane Smith	Pending
99	Meeting 99	Meeting 99 Description	Office 99	2025-10-31	13:00:00	14:00:00	John Doe, Jane Smith	Pending
100	Meeting 100	Meeting 100 Description	Office 100	2025-11-01	09:00:00	10:00:00	John Doe, Jane Smith	Pending

SQL Before Meeting

ID	Title	Description	Location	Date	Start Time	End Time	Attendees	Status
1	Meeting 1	Meeting 1 Description	Office 1	2025-07-25	10:00:00	11:00:00	John Doe, Jane Smith	Pending
2	Meeting 2	Meeting 2 Description	Office 2	2025-07-26	14:00:00	15:00:00	John Doe, Jane Smith	Pending
3	Meeting 3	Meeting 3 Description	Office 3	2025-07-27	09:00:00	10:00:00	John Doe, Jane Smith	Pending
4	Meeting 4	Meeting 4 Description	Office 4	2025-07-28	13:00:00	14:00:00	John Doe, Jane Smith	Pending
5	Meeting 5	Meeting 5 Description	Office 5	2025-07-29	08:00:00	09:00:00	John Doe, Jane Smith	Pending
6	Meeting 6	Meeting 6 Description	Office 6	2025-07-30	12:00:00	13:00:00	John Doe, Jane Smith	Pending
7	Meeting 7	Meeting 7 Description	Office 7	2025-07-31	15:00:00	16:00:00	John Doe, Jane Smith	Pending
8								



SQL After Meeting



SQL After Attendees



☞ Task #3 (0.30 pts.) - Links

Progress: 100%

Details:

- Include the heroku prod link to this page (even though it's not a full page, it's more like how logout works)
 - Include pull request link for this feature

URL #1

<https://ng569-it202-450->



<https://ng569-it202-450-prod>



prod-3272507b1c51.herokuapp.com/project/edit_meetings.php

URL #2

<https://github.com/Nate-Gaw/ng569-IT202-450/pull/44>



[URL](https://github.com/Nate-Gaw)



Section #7: (0.5 pts.) Misc

Progress: 100%

≡ Task #1 (0.17 pts.) - Github Details

Progress: 100%

▣ Part 1:

Progress: 100%

Details:

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



Github Commits



Saved: 7/25/2025 7:56:25 PM

◐ Part 2:

Progress: 100%

Details:

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

URL #1

<https://github.com/Nate-Gaw/ng569->

IT20241504



URL

<https://github.com/Nate-Gaw/ng5>



Saved: 7/25/2025 7:56:25 PM

▣ Task #2 (0.17 pts.) - WakaTime - Activity

Progress: 100%

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

Projects · ng569-IT202-450



22 hrs 56 mins over the Last 7 Days in ng569-IT202-450 under all branches.

wakatime

```
git clone https://github.com/wakatime/wakatime.git
cd wakatime
make install
# or
curl -s https://wakatime.com/api/v1/install | bash
```

wakatime individual time

```
git clone https://github.com/wakatime/wakatime.git
cd wakatime
make install
# or
curl -s https://wakatime.com/api/v1/install | bash
```

wakatime individual time

Saved: 7/25/2025 7:58:37 PM

☰ Task #3 (0.17 pts.) - Reflection

Progress: 100%

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

Progress: 100%

Details:

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

Your Response:

I learned a lot using bootstrap. I really like how easy it was to make things look prettier. I also learned a lot about how POST vs GET and how submitting data works between various languages. Its honestly really cool to me, and I feel like im actually programming something meaningful, not just "hello world" for the 900th time. Also experimenting with some of the database options was really cool to learn and figure out. I had to watch a 3 hour tutorial to figure out how sql works and how to program in it. But afterwards, I was able to make some pretty cool things out of it. I also learned a bit more of JS. I really don't like JS because of how difficult it is to read and process for me. Unlike python, html or even php, JS is hard to understand by reading it. But I did have to for filtering, so im glad I figured it out.



Saved: 7/25/2025 8:03:38 PM

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

Progress: 100%

Details:

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

Your Response:

Easiest part of the assignment was probably the tables and landing page. SQL is a very easy language to read and understand so doing a bunch of calls to get data was easy in my opinion. Most of the assignment was fairly easy, just needed a lot of time to thin out the herd of information. Another easy part was the HTML (NOT CSS). I mapped out all my pages on my tablet, which allowed me to see what I wanted on each page (even if it's not the prettiest yet). Furthermore, html was pretty self-explanatory to me, my only issue being that there are too many differnet things I can do with html.



Saved: 7/25/2025 8:08:59 PM

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

Progress: 100%

Details:

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

Your Response:

Hardest part of the assignment was definitely anything with JS. Whether it was my filtering or JS validation, trying to remember and keep up with the key words was a pain. The majority of my time on my pages and on the landing was figuring out the JS filtering system and JS validation. Also, figuring out how to make the edit page was a pain because I was trying to make it similar to the create age or even reuse the edit page, but I couldn't just because of how cluttered and messed up things would get. I would definitely say its possible but the code is gonna be thousands of lines by the end.



Saved: 7/25/2025 8:07:25 PM