Submission Worksheet

Submission Data

Course: IT202-450-M2025
Assignment: IT202 - Milestone 3
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: 8/1/2025 3:10:32 PM
Updated: 8/1/2025 4:33:27 PM
Grading Link: https://learn.ethereallab.app/assignment/v3/IT202-450-M2025/it202-milestone-3/grading/ng569
View Link: https://learn.ethereallab.app/assignment/v3/IT202-450-M2025/it202-milestone-3/view/ng569

Instructions

1. Refer to Milestone3 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 Milestone3 branch
    1. `git checkout Milestone3` (ensure proper starting branch)
    2. `git pull origin Milestone3` (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 Milestone3`
    4. On Github merge the pull request from `Milestone3` 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: (3 pts.) Api Data

Progress: 100%

Task #1 (1 pt.) - Concept of Data Association

Progress: 100%

Details:

- What's the concept of your data association to users? (examples: favorites, wish list, purchases, assignment, etc)
- Describe with a few sentences

Your Response:

The data I grab from the Api is the location of the use. So I converted times into the users appropriate timezone. This allows the user to be able to have an easier time using the UI and will vary per person's location. Furthermore, if the user is not an admin, the user wil only be able to see their meetings that they are a part of and not other meetings they are not a part of.

Saved: 8/1/2025 3:12:31 PM

Task #2 (1 pt.) - Data Updates

Progress: 100%

Details:

- When an associated entity is updated (manually or API) how is the association affected?
  - Does the user see the old version of the data?
  - Does the user see the new version of the data?
  - Does the user need to have data re-associated or remapped?
- Explain why.

Your Response:

In my case, when a meeting is updated, the previous information is lost and when the user reloads the page, the new information will be overlayed. The user doesn't need to do anything with the new data, as the backend php will update everything automatically. So if the user changes locations, the user will just need to update the profile information and all the meeting data and times will be changed to the updated location. This also applies for creating new meetings. When the user changes their location and creates a new meeting, the meeting will be according to the new location's time, not the old.

Saved: 8/1/2025 3:15:09 PM

Task #3 (1 pt.) - Handling Association

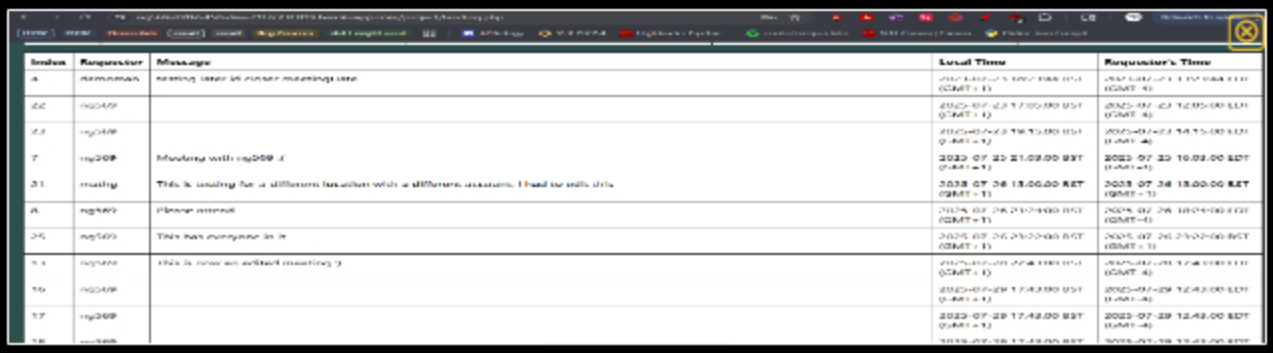
Progress: 100%

Part 1:

Progress: 100%

Details:

- Show an example page of where a user can get data associated with them
- Ensure heroku dev url is visible
- Caption if this is a user-facing page or admin page



Index	Requester	Message	Local Time	Requester's Time
4	demoman	meeting later in closer meeting one	[illegible] (GMT + 1)	[illegible] (GMT + 4)
22	ng049		[illegible] (GMT + 1)	[illegible] (GMT + 4)
23	ng049		[illegible] (GMT + 1)	[illegible] (GMT + 4)
7	ng069	Meeting with ng069 :/	2023-07-29 21:00:00 BST (GMT + 1)	2023-07-29 16:00:00 EDT (GMT + 4)
31	meeting	This is meeting for a different location with a different usersm. I had to edit this	2023-07-30 13:00:00 BST (GMT + 1)	2023-07-30 13:00:00 BST (GMT + 1)
8	ng069	Please attend	[illegible] (GMT + 1)	[illegible] (GMT + 1)
25	ng069	This has everyone in it	[illegible] (GMT + 1)	[illegible] (GMT + 1)
14	ng049	This is now an edited meeting :)	[illegible] (GMT + 1)	[illegible] (GMT + 4)
16	ng049		[illegible] (GMT + 1)	[illegible] (GMT + 4)
17	ng069		2023-07-29 17:43:00 BST (GMT + 1)	2023-07-29 12:43:00 EDT (GMT + 4)

This shows the landing page, where the times are converted based on where the user's set location is, this is a user-facing.

Saved: 8/1/2025 3:24:32 PM

Part 2:

Progress: 100%

Details:

- Describe the process of associating data with the user.
- Can it be toggled, or is it applied once?

Your Response:

When the user registers for the webpage, they must input their approximate location, which will associate the api's return data with the user. It can only be changed in the profile page, and all data and times will be converted to the users's inputted location. Futhermore, when other users are associated with the same data, the code converts the time into the other user's timezone, which allows all users to easily see appropriate meeting times, without the need to convert anything.

Saved: 8/1/2025 3:24:32 PM

Section #2: (6 pts.) Associations

Progress: 100%

Task #1 (1.50 pts.) - Logged-in User's Associated Entities

Progress: 100%

Details:

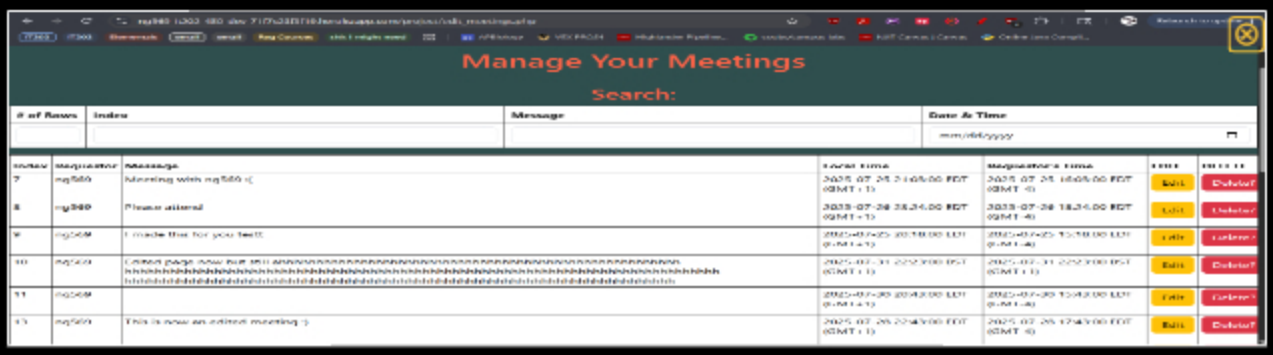
- Each line item should have a logical summary
- Each line item should have a link/button to a single view page of the entity
- Each line item should have a link/button to a delete action of the relationship (doesn't delete the entity or user, just the relationship)
- The page should have a link/button to remove all associations for the particular user
- The page should have a section for stats (number of results and total number possible based on the query filters)
- The page should have logical options for filtering/sorting
  - A limit should be applied between 1 and 100 and controlled by the user (server-side enforces rules)
  - A filter with no matching records should show "no results available" or equivalent

Part 1:

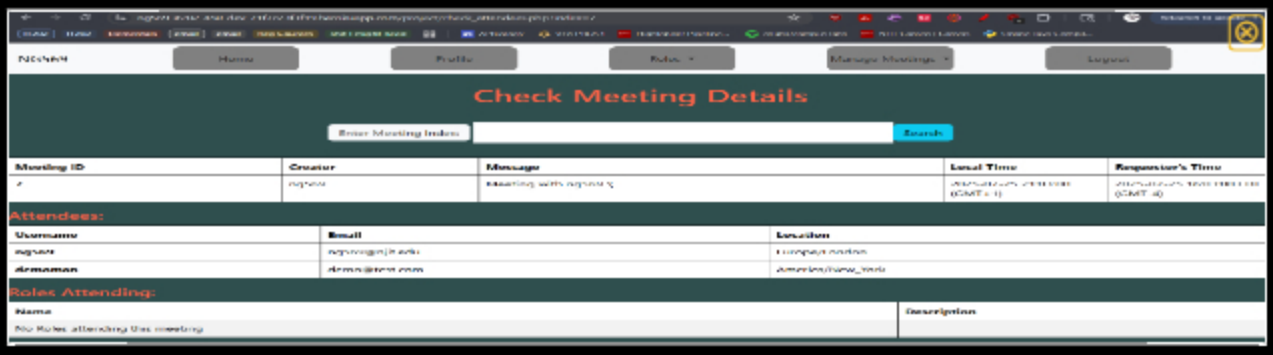
Progress: 100%

Details:

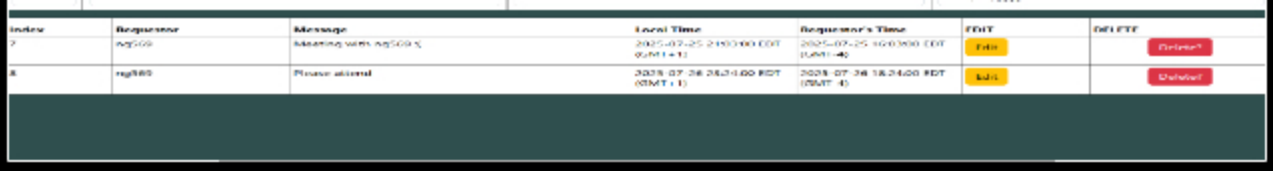
- Show a few examples of this page from heroku dev with various filters applied
- Ensure heroku dev url is visible
- Ensure each requirement is visible



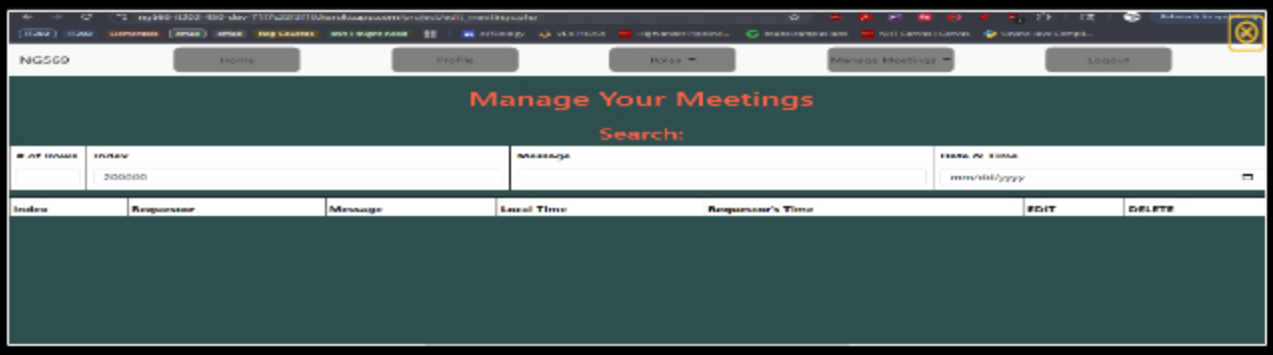
Default edit page with search bar



Summary page for meeting details to reduce clutter on previous page



Showing seach row



empty search results

Saved: 8/1/2025 4:08:16 PM

Part 2:

Progress: 100%

Details:

- Describe how you solved showing the particular association output
- Describe how you solved the various items required for this page (i.e., line item requirements, stats, filter/sort, etc)

Your Response:

The pictures show the meetings the user has created and is able to modify their own meetings with an edit and delete button. I was able to do this by filtering in teh DB call by only grabbing data that has the users id located in it. The user is also able to click on the table row in order to get a full detail about the meeting, attendees, etc. The search bar also updates the page, using JS, without the need to reload or buffer. I didn't want a confirmation alert if there was no results since the page updates automatically to reduce the clutter on the page. In my opinion, there is already a lot on the page in terms of meeting, information and a seach bar, so I wanted to reduce the amount on the page, and move excess infomation to other pages.

Saved: 8/1/2025 4:08:16 PM

Task #2 (1.50 pts.) - All Users Association Page

Progress: 100%

Details:

- Each line item should have a logical summary
- Each line item should include the username this entity is associated with

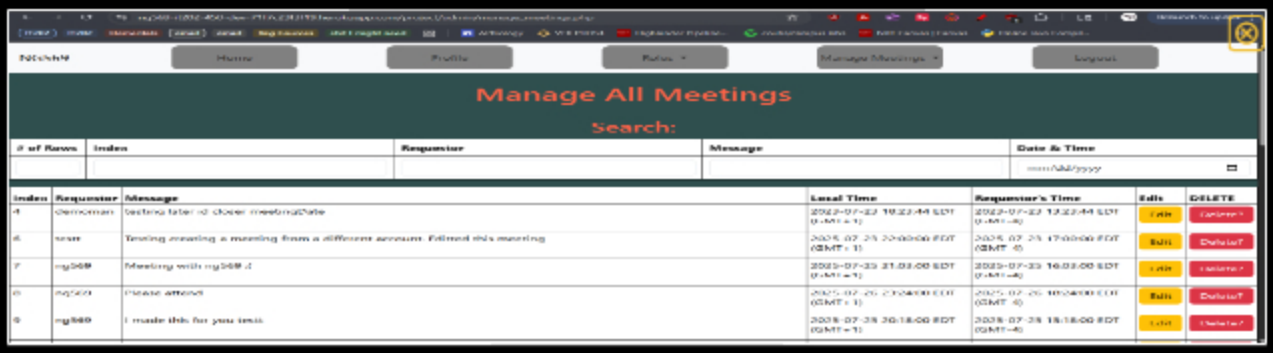
    - Clicking the username should redirect to that user's public profile
- Each line item should include a column that shows the total number of users the entity is associated with
- Each line item should have a link/button to a single view page of the entity
- Each line item should have a link/button to a delete action of the relationship (doesn't delete the entity or user, just the relationship)
- The page should have a section for stats (number of results and total number possible based on the query filters)
- The page should have logical options for filtering/sorting
    - A limit should be applied between 1 and 100 and controlled by the user (server-side enforces rules)
    - A filter with no matching records should show "no results available" or equivalent

Part 1:

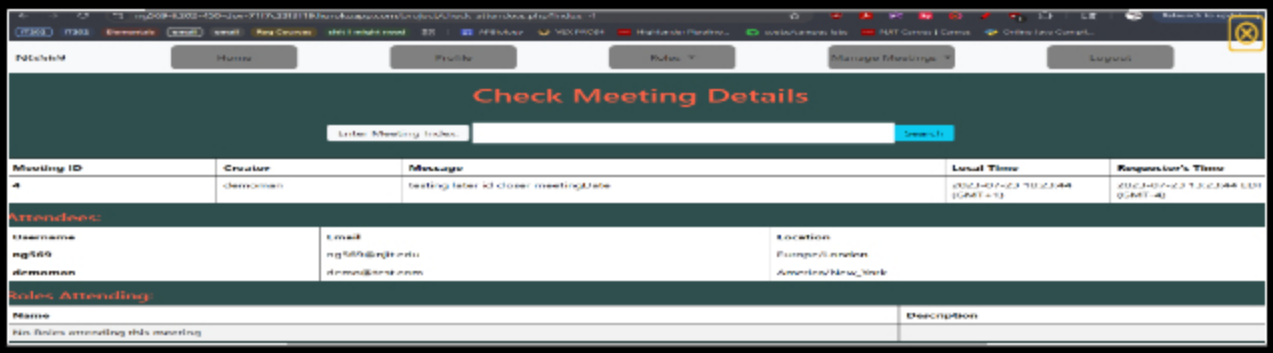
Progress: 100%

Details:

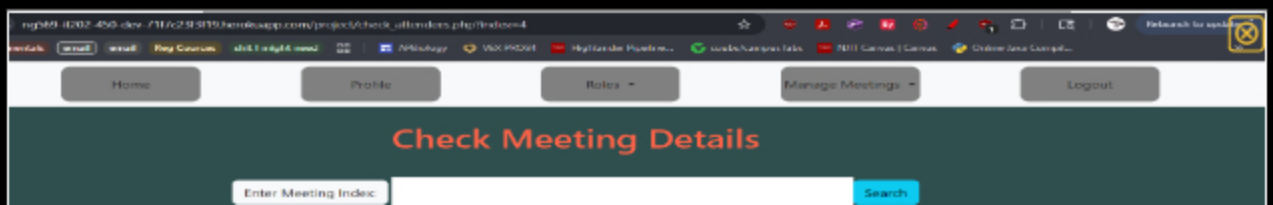
- Show a few examples of this page from heroku dev with various filters applied
- Ensure heroku dev url is visible
- Ensure each requirement is visible



Shows every meeting and who created each meeting (admin page)



Shows details of any meeting (admin page0



Creator	Message	Local Time	Requestor's Time
demoman	testing later id closer meetingDate	2023-07-23 18:23:44 (GMT+1)	2023-07-23 13:23:44 EDT (GMT-4)

Email	Location

There isn't really a public profile, although showing details shows the time for each user in their respective lcoation

Saved: 8/1/2025 4:09:56 PM

Part 2:

Progress: 100%

Details:

- Describe how you solved showing the particular association output
- Describe how you solved the various items required for this page (i.e., line item requirements, stats, filter/sort, etc)

Your Response:

I was able to create this page by creating a general DB call and grabbing all the information possible. Then I was able to use the same search bar from the previous section in order to update everything on the page automatically using JS. This works very similar to the previous section, except this outputs everyones meeting, to allow people with only the admin role to manage, delete or edit any meetings, in case something goes wrong.

Saved: 8/1/2025 4:09:56 PM

Task #3 (1.50 pts.) - Unassociated Page

Progress: 100%

Details:

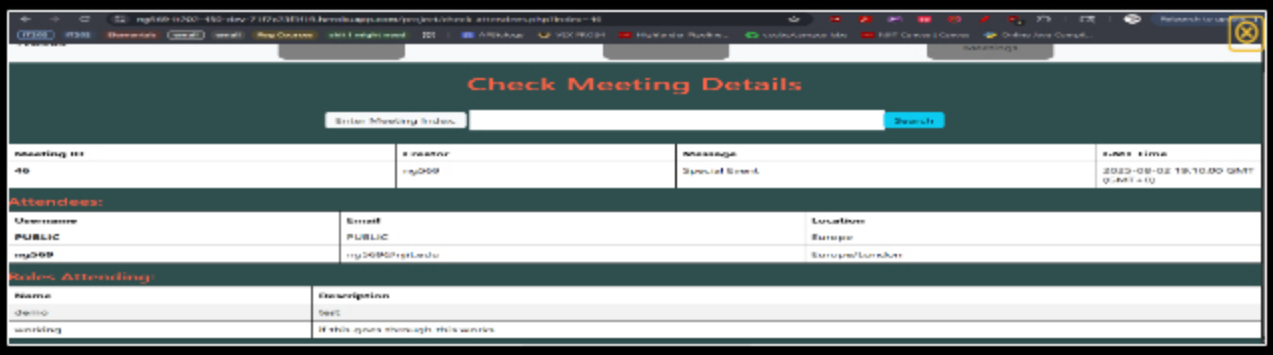
- Each line item should have a logical summary
- Each line item should have a link/button to a single view page of the entity
- The page should have a section for stats (number of results and total number possible based on the query filters)
- The page should have logical options for filtering/sorting
  - A limit should be applied between 1 and 100 and controlled by the user (server-side enforces rules)
  - A filter with no matching records should show "no results available" or equivalent

Part 1:

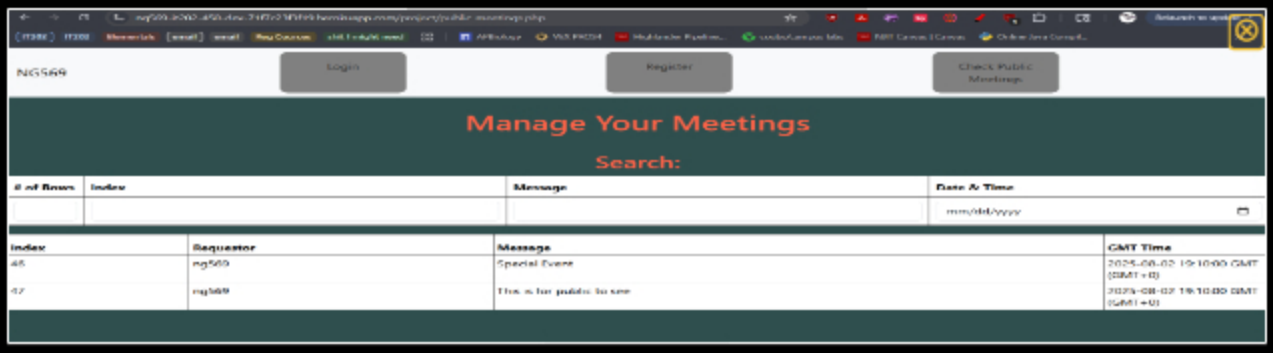
Progress: 100%

Details:

- Show a few examples of this page from heroku dev with various filters applied
- Ensure heroku dev url is visible
- Ensure each requirement is visible



Public detail page (accessed not logged in)



Unassociated page (public page)

Saved: 8/1/2025 4:17:39 PM

Part 2:

Progress: 100%

Details:

- Describe how you solved showing the particular association output
- Describe how you solved the various items required for this page (i.e., line item requirements, stats, filter/sort, etc)

Your Response:

This page works by only grabbing meeting information where the public is associated with the meeting. The seach bar is the same JS auto update search bar. This is a page where if a business wants to create meetings for the public or meetings for public knoledge, they are able to create a page and add a user called PUBLIC in it. This then adds the user PUBLIC as normal and treats them as another user in the datatbase. This alos goes for security, where they are given a -1 as their ID, so if they try to access another index meeting or page, their id will come back as a -1, denying them.

Saved: 8/1/2025 4:17:39 PM

☰ Task #4 (1.50 pts.) - Admin Association Page (Like User Roles)

Progress: 100%

Details:

- The page should have a form with two fields
  - Partial match for username
  - Partial match for entity reference (name or something user-friendly)
- Submitting the form should give up to 25 matches of each
  - Likely best to show as two separate columns
- Each entity and user will have a checkbox next to them
- Submitting the checked associations should apply the association if it doesn't exist; otherwise it should remove the association
  - A filter with no matching records should show "no results available" or equivalent

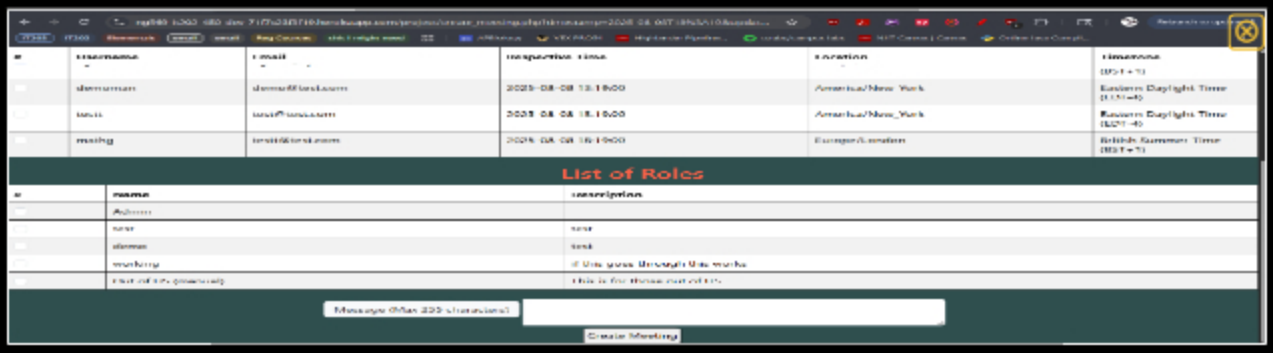
🖼 Part 1:

Progress: 100%

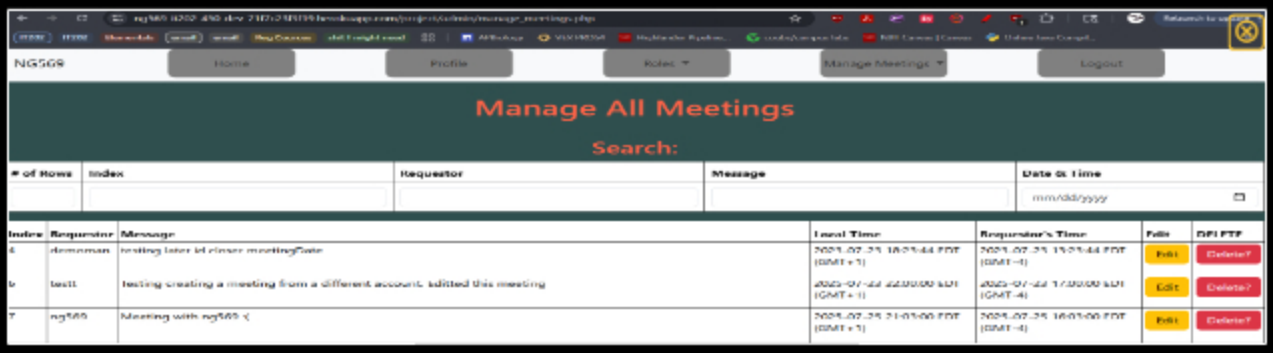
Details:

- Show a few examples of this page from heroku dev with various selections having been submitted
- Ensure heroku dev url is visible
- Ensure each requirement is visible

Adding users to meetings



Adding roles to meetings



Admin view all meetings page

Saved: 8/1/2025 4:25:23 PM

Part 2:

Progress: 100%

Details:

- Describe how your code solves the requirements; be clear

Your Response:

So anyone can actually create an association, since I wanted everyone, who logs in, to be able to create a meeting when needed. But, the admins are able to edit and delete ANY meetings, whereas users are only able to change and see their own created meetings. This is so if someone is creating a bunch of random meetings, or if someone makes a mistake making a meeting and needs an admin to change it, they are able to see all the information of who makes the meetings, times, and able to modify or delete the meetings. Futhermore, I added the ability for anyone to create a meeting and add roles to the meeting, so anyone who was part of that role will be able to see the correct time and the meeting in general.

Saved: 8/1/2025 4:25:23 PM

Section #3: (1 pt.) Misc

Progress: 100%

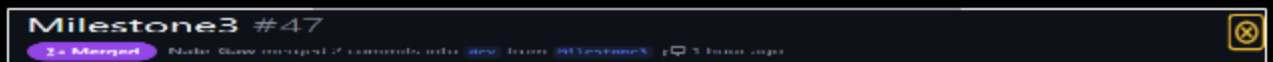
Task #1 (0.33 pts.) - Github Details

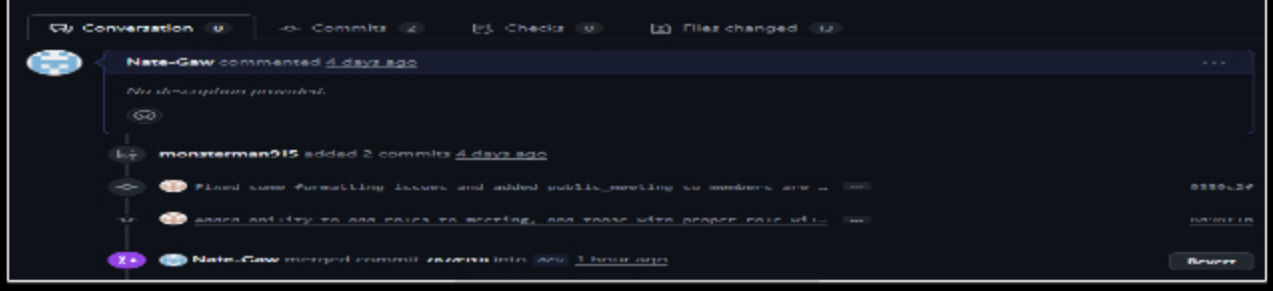
Progress: 100%

Part 1:

Progress: 100%

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





Github Commits

Saved: 8/1/2025 4:26:25 PM

Part 2:

Progress: 100%

Details:
Include the link to the Pull Request for Milestone3 to dev (should end in `/pull/#`)

URL #1
https://github.com/Nate-Gaw/ng569-IT202-450/pull/7

URL
https://github.com/Nate-Gaw/ng5

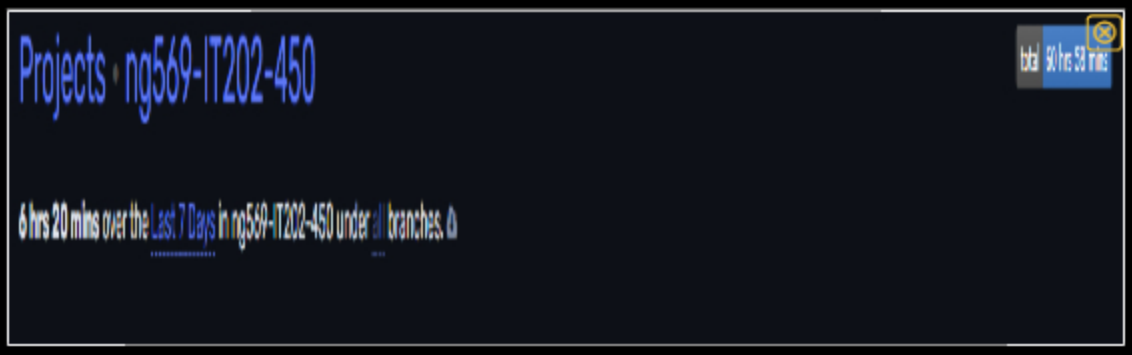
Saved: 8/1/2025 4:26:25 PM

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

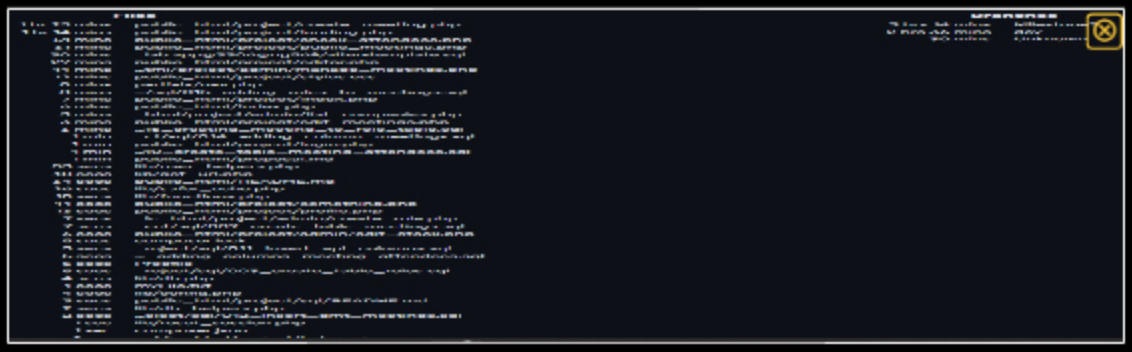
Progress: 100%

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



Wakatime total time



Wakatime Files

Saved: 8/1/2025 4:27:35 PM

Task #3 (0.33 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 about handling a lot of DB information in SQL. Joining and filtering the databases was pretty tiresome, and sometimes I would end up leaving out some information or double entries would show up (especially with the new addition of including roles in meetings). Fortunately I was able to get though it, but a lot of doubling would show up and sometimes I would sort that out through php, since the Sql was getting long and complicated already.

Saved: 8/1/2025 4:29:36 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 milestone 3 was the html. I didn't run into many problems with it and most of my pages were laid out the same, so it was kind of copy and paste

and then have fun with the CSS and formatting. Also, the HTML wasn't difficult either, since I started using Bootstrap for most of it; it just made the CSS harder, trying to override some of the Bootstrap.

Saved: 8/1/2025 4:31:18 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 adding the ability to add roles into attenddees for meetings. In my mind I wanted to add another column for roles, but then I ran into a bunch of issues trying to update the old information into handling the new (which seems to be a problem in every project), so I decided to just create a new table to handle all of the role and meeting information since I could essentially copy the origional attendee_meetings table and just put roles in. This would come at the cost of having more SQL calls, but I did my best to limit it to 1 or 2 more calls most.

Saved: 8/1/2025 4:33:27 PM