Submission Worksheet

Submission Data

Course: IT202-450-M2025
Assignment: IT202 Milestone 1
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: 7/7/2025 1:25:56 AM
Updated: 7/7/2025 10:13:57 PM
Grading Link: https://learn.ethereallab.app/assignment/v3/IT202-450-M2025/it202-milestone-1/grading/ng569
View Link: https://learn.ethereallab.app/assignment/v3/IT202-450-M2025/it202-milestone-1/view/ng569

Instructions

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

1. Refer to Milestone1 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 Milestone1 branch
    1. `git checkout Milestone1` (ensure proper starting branch)
    2. `git pull origin Milestone1` (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 Milestone1`
    4. On Github merge the pull request from `Milestone1` 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.) Feature: User Will Be Able To Register A New Account

Progress: 100%

Progress: 100%

Task #1 (0.67 pts.) - Validation

Progress: 100%

Details:

- Show the relevant code snippets for each validation layer
- Show the relevant demo of each validation layer
- Briefly explain how the validation steps work at each layer

Task #1 (0.33 pts.) - HTML Form/Validation

Progress: 100%

Details:

- System should allow the user to correct the error without wiping/clearing the form (for the PHP side this includes sticky-form logic to keep the username/email address entries)

Part 1:

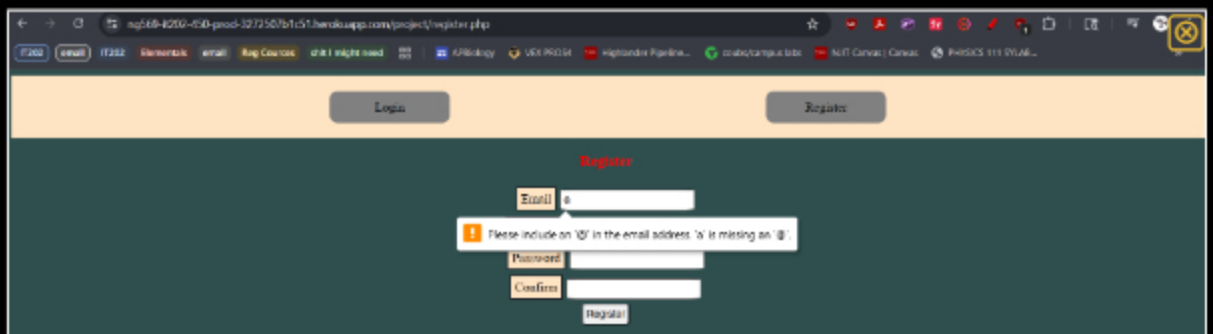
Progress: 100%

Details:

- Show the code related to the register page's form (HTML validation)
- Show examples of each validation message (you may be able to capture this in one or few screenshots)

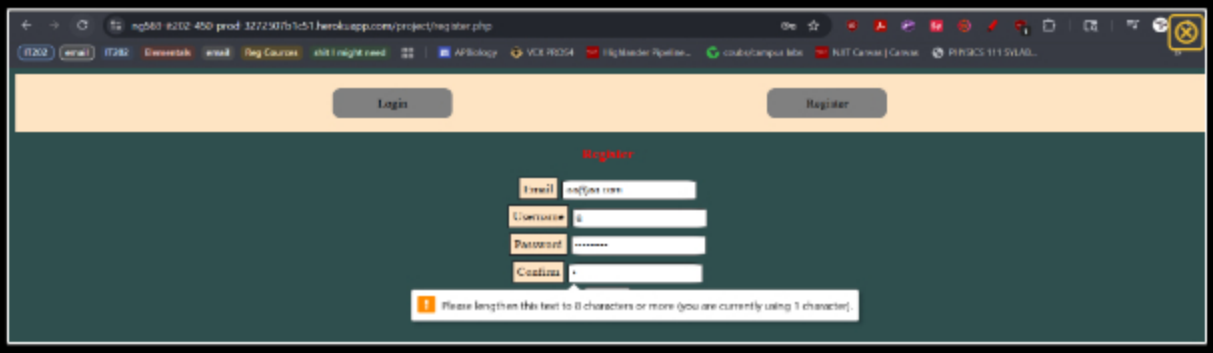
```
<form onsubmit="return validate(this)" method="POST">
    <!--ng569 ////25
    HTML  form with email username password inputs and a confirm button-->
    <div>
        <label for="email">Email</label>
        <input id="email" type="email" name="email" required />
    </div>
    <div>
        <label for="username">Username</label>
        <input type="text" name="username" required maxlength="30" />
    </div>
    <div>
        <label for="pw">Password</label>
        <input type="password" id="pw" name="password" required minlength="8" />
    </div>
    <div>
        <label for="confirm">Confirm</label>
        <input type="password" name="confirm" required minlength="8" />
    </div>
    <input type="submit" value="Register" />
</form>
```

Code for Register



Email Validation

Password Validation



Confirm Validation

Saved: 7/7/2025 1:44:28 AM

Part 2:

Progress: 100%

Details:

- Briefly explain the validation step (i.e, what you chose, how they solve the requirement)

Your Response:

The html validation was taken from the lectures. Using html type, the email and passwords are checked for length and formatted according to the html types.

Saved: 7/7/2025 1:44:28 AM

Task #2 (0.33 pts.) - JS Validation (validate() function)

Progress: 100%

Part 1:

Progress: 100%

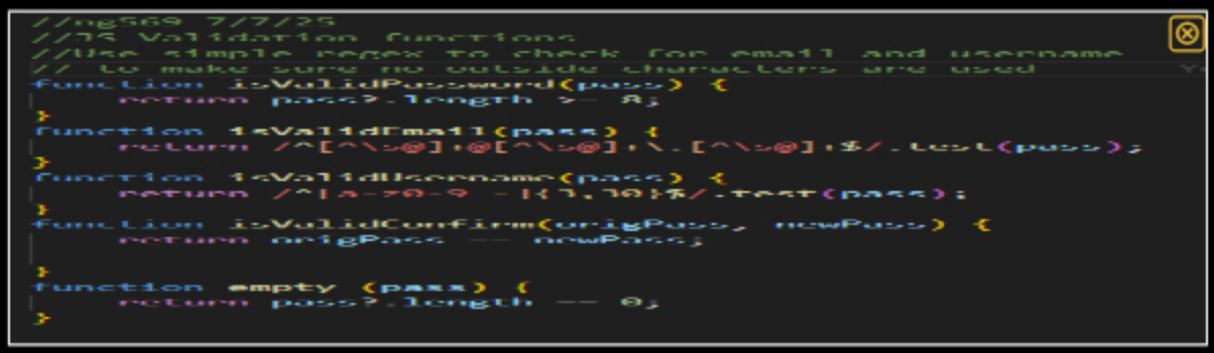
Details:

Details:

- Show the code related to the register page's validate() function
- Show examples of each validation message [username format, email format, password format, password doesn't match confirm](you may be able to capture this in one or few screenshots)
- Note: You may need to use dev tools to temporarily apply `novalidate` to the form tag

JS Validation Code

Heroku Validation



JS Validation functions

Saved: 7/7/2025 2:09:46 AM

Part 2:

Progress: 100%

Details:

- Briefly explain the validation step (i.e, what you chose, how they solve the requirement)

Your Response:

I used {regex}.test in order to compare the string to a regex expression to prove the validation of the email and username. Then by using string?.length I am able to do a one-line return statement and check the length of the string for validation.

Saved: 7/7/2025 2:09:46 AM

Task #3 (0.33 pts.) - PHP Validation (steps before the DB call)

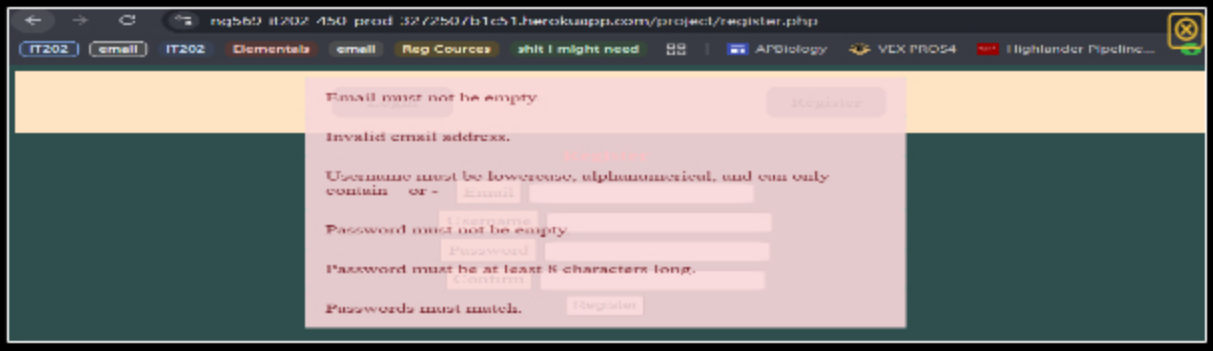
Progress: 100%

Part 1:

Progress: 100%

Details:

- Show the code related to the register page's PHP validation
- Show examples of each validation message [username format, email format, password format, invalid password, password doesn't match confirm, username taken, email taken] (you may be able to capture this in one or few screenshots)
- Note: You may need to use dev tools to temporarily apply `novalidate` to the form tag and temporarily override the validate() function or use the provided helper script in the instructions
- Include the outputs related to username/email already in use



Heroku Validation

PHP Validation Code

```php
    if (!$hasError) {
        // TODO 4: Hash password and store record in DB
        $hashed_password = password_hash($password, PASSWORD_BCRYPT);
        $db = getDB(); // available due to the 'require()' of 'functions.php'
        // Code for inserting user data into the database
        $stmt = $db->prepare("INSERT INTO Users (email, password, username) VALUES (:email, :password, :username)");
        try {
            $stmt->execute([':email' -> $email, ':password' -> $hashed_password, ':username' -> $username]);

            flash("Successfully registered! You can now log in.", "success");
        } catch (PDOException $e) {
            // Handle duplicate email/username
            users_check_duplicate($e);
        } catch (Exception $e) {
            flash("There was an error registering. Please try again.", "danger");
            error_log("Registration Error: " . var_export($e, true)); // log the technical error for debugging
        }
    } <- #110-122// if (!$hasError)
```

PHP Validation Code 2

```php
<?php
//ngSG0 7/7/25
//PHP Functions, using filter_var functions to sanitize and validate traditional email syntax
//then uses preg_match function to check for valid username
function sanitize_email($email = "") {
    return filter_var(trim($email), FILTER_SANITIZE_EMAIL);
}
function is_valid_email($email = "") {
    return filter_var(trim($email), FILTER_VALIDATE_EMAIL);
}
function is_valid_username($username)
{
    return preg_match('/^[a-z0-9_]{3,30}$/', $username);
}
function is_valid_password($password)
{
    return strlen($password) >= 8;
}
function is_valid_confirm($original, $confirm)
{
    // checking not empty to avoid empty equals empty being true
    return !empty($original) && $original === $confirm;
}
```

PHP Functions

Saved: 7/7/2025 2:15:30 AM

Part 2:

Progress: 100%

Details:

- Briefly explain the validation step (i.e, what you chose, how they solve the requirement)

Your Response:

I used filter_var in order to sanitize and validate traditional emails. Then preg-match is used to ensure the username is correctly validated and put together.

Saved: 7/7/2025 2:15:30 AM

Task #2 (0.67 pts.) - Related URLs

Progress: 100%

Details:

- Include the direct link to this file from the Milestone branch (should end in .php)
- Include the heroku prod link to this file (Just grab the base prod url and

manually enter the path to the file)
- Include the related pull request link for this feature

URL #1
https://github.com/Nate-Gaw/ng569-IT202-450/Milestone1/public_html/project/register.php
URL
https://github.com/Nate-Gaw

URL #2
https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/register.php
URL
https://ng569-it202-450-prod

URL #3
https://github.com/Nate-Gaw/ng569-IT202-450/pull/25
URL
https://github.com/Nate-Gaw

URL #4
https://github.com/Nate-Gaw/ng569-IT202-450/pull/24
URL
https://github.com/Nate-Gaw

URL #5
https://github.com/Nate-Gaw/ng569-IT202-450/pull/21
URL
https://github.com/Nate-Gaw

URL #6
https://github.com/Nate-Gaw/ng569-IT202-450/pull/20
URL
https://github.com/Nate-Gaw

URL #7
https://github.com/Nate-Gaw/ng569-IT202-450/pull/20
URL
https://github.com/Nate-Gaw

URL #8
https://github.com/Nate-Gaw/ng569-IT202-450/pull/19
URL
https://github.com/Nate-Gaw

URL #9
https://github.com/Nate-Gaw/ng569-IT202-450/pull/18
URL
https://github.com/Nate-Gaw

URL #10
https://github.com/Nate-Gaw/ng569-IT202-450/pull/11
URL
https://github.com/Nate-Gaw

Saved: 7/7/2025 5:41:19 PM

Task #3 (0.67 pts.) - Database - Users table

Progress: 100%

Details:

- Add a screenshot of the database view from vs code showing a valid registered user (preferrably a recent one during evidence capturing)

id int	email varchar(100)	password varchar(60)	created timestamp	modified timestamp	username varchar(30)
1	ng569@njit.edu	$2y$12$FLtLBhRYwb86toO7	2025-06-20 02:41:24	2025-06-20 02:41:24	ng569-d399cbe974894875(
4	demo@test.com	$2y$12$J6deU42lII9StPxXrP.	2025-06-28 01:23:50	2025-07-06 23:17:58	demoman
6	test@test.com	$2y$12$unhxnqjYwKmFY3q	2025-07-07 06:32:46	2025-07-07 06:32:46	testt

VS Code SQL DB

Saved: 7/7/2025 2:34:29 AM

Section #2: (2 pts.) Feature: User Will Be Able To Login To Their Account

Progress: 100%

Task #1 (0.67 pts.) - Validation

Progress: 100%

Details:

- Show the relevant code snippets for each validation layer
- Show the relevant demo of each validation layer
- Briefly explain how the validation steps work at each layer

Task #1 (0.33 pts.) - HTML Form/Validation

Progress: 100%

Details:

- System should allow the user to correct the error without wiping/clearing the form (for the PHP side this includes sticky-form logic to keep the username/email address entries)

Part 1:

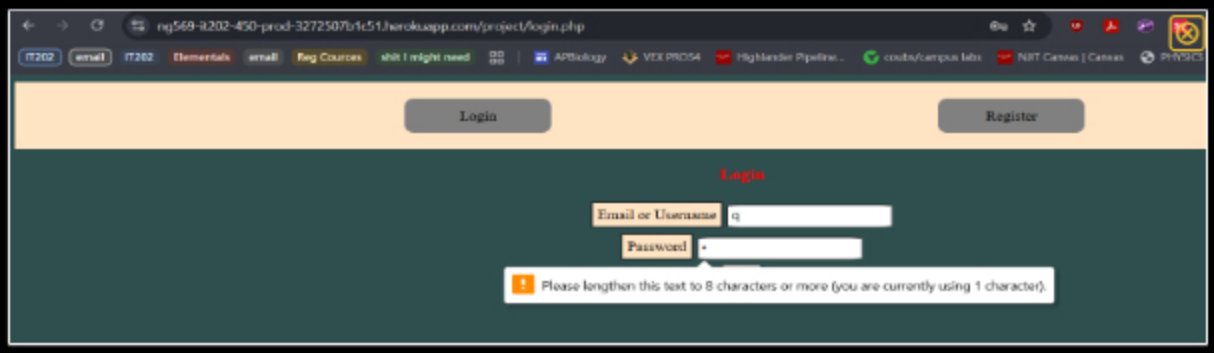
Progress: 100%

Details:

- Show the code related to the login page's form (HTML validation)
- Show examples of each validation message (you may be able to capture this in one or few screenshots)

```
<form onsubmit="return validate(this)" method="POST">
    <!--ng569 7/725
    HTML Form, 2 inputs 1 for email and password and 1 button to submit the form-->
    <div>
        <label for="email">Email or Username</label>
        <input id="email" type="text" name="email" required />
    </div>
    <div>       You, 3 weeks ago • Added Nav and Login Files …
        <label for="pw">Password</label>
        <input type="password" id="pw" name="password" required minlength="8" />
    </div>
    <input type="submit" value="Login" />
</form>
```

HTML Form Code



Heroku HTML Validation

Saved: 7/7/2025 4:52:18 PM

Part 2:

Progress: 100%

Details:

- Briefly explain the validation step (i.e, what you chose, how they solve the requirement)

Your Response:

There is only validation for password, as even if the user doesn't input an email format, they could be using a username to login. So only password is validated to check for a min length of 8.

Saved: 7/7/2025 4:52:18 PM

Task #2 (0.33 pts.) - JS Validation (validate() function)

Progress: 100%

Part 1:

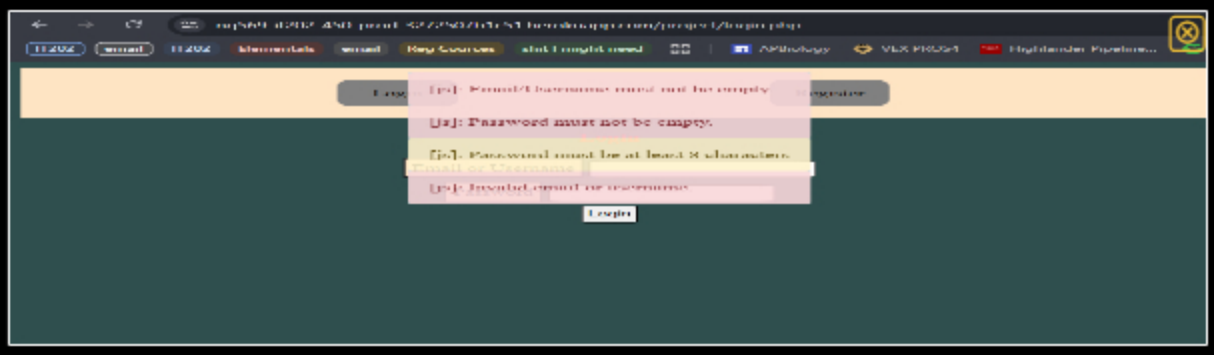
Progress: 100%

Details:

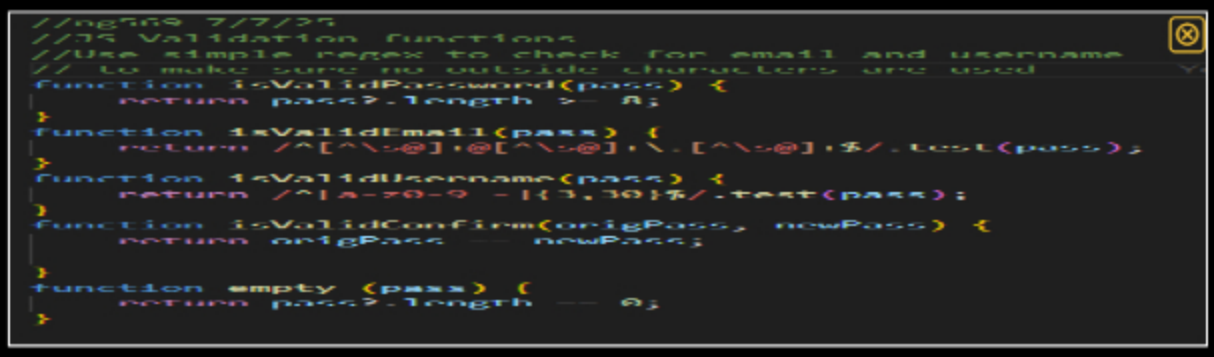
- Show the code related to the login page's validate() function

- Show examples of each validation message [username format, email format, password format] (you may be able to capture this in one or few screenshots)
- Note: You may need to use dev tools to temporarily apply `novalidate` to the form tag

JavaScript Code



Heroku Javascript Validation



Javascript Functions

Saved: 7/7/2025 4:59:53 PM

Part 2:

Progress: 100%

Details:

- Briefly explain the validation step (i.e, what you chose, how they solve the requirement)

Your Response:

My code checks to see if both inputs are empty. If they are then the code will flag the issue. If not then the password is checked for length <8. After the email/username input is checked to see if the input is a valid email or valid username, the site then outputs the same code if the input is neither an email nor a username. Since password, email and username requirements are all the same, the functions are reused from the registration site.

Saved: 7/7/2025 4:59:53 PM

Task #3 (0.33 pts.) - PHP Validation (steps before the DB call)

Progress: 100%

Part 1:

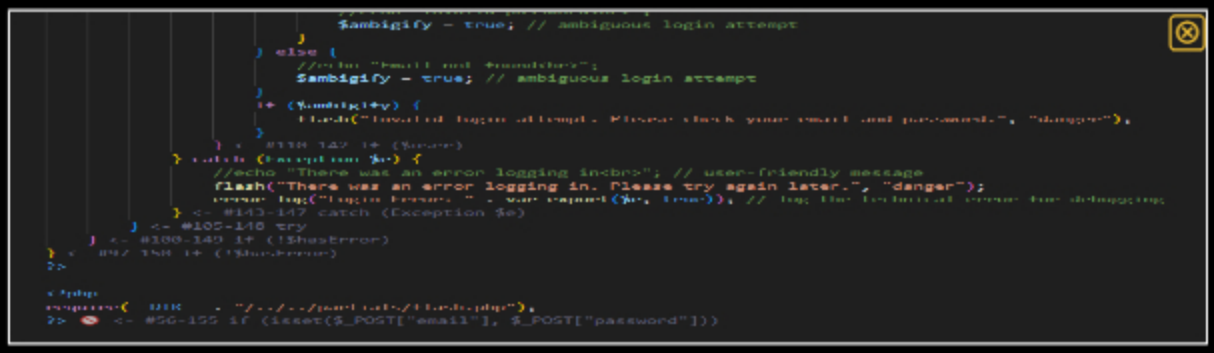
Progress: 100%

Details:

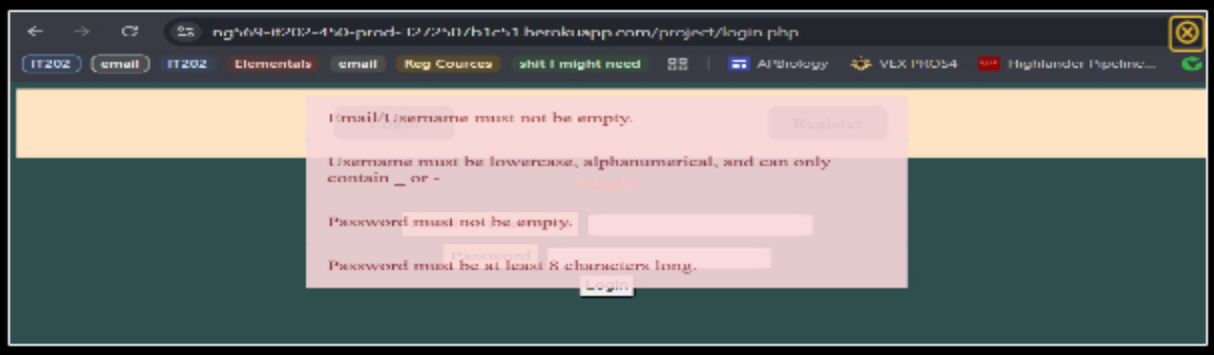
- Show the code related to the login page's PHP validation
- Show examples of each validation message [username format, email format, password format, invalid password] (you may be able to capture this in one or few screenshots)
- Note: You may need to use dev tools to temporarily apply `novalidate` to the form tag and temporarily override the validate() function or use the provided helper script in the instructions
- Include the outputs related to user doesn't exist and php-side password mismatch

Login PHP Code

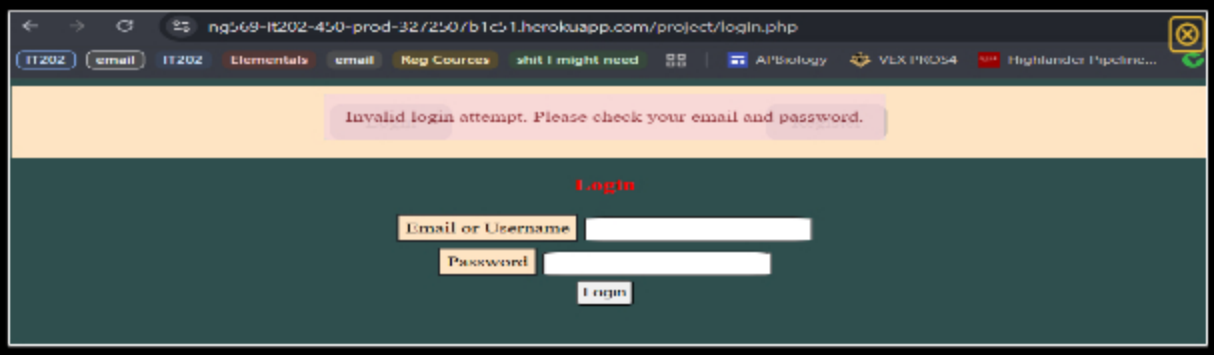
Login PHP Code 2



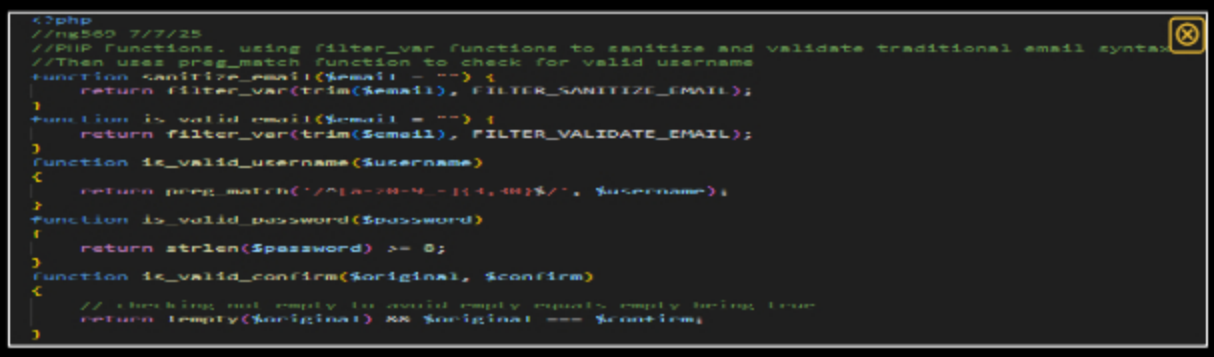
Login PHP Code 2



Heroku PHP Output



Heroku Login Failed Attempt



PHP Functions

Saved: 7/7/2025 5:13:28 PM

Part 2:

Progress: 100%

Details:

- Briefly explain the validation step (i.e, what you chose, how they solve the requirement)

Your Response:

Since most of the validation is similar to other pages, the functions were reused from previous pages. Then SQL commands were used in order to retrieve the usernames and passwords from the User database, and potential roles from the UserRoles database in order to look up and prepare for potential roles needed inside the landing page. But if nothing fits, then an alert is issued to the user.

Saved: 7/7/2025 5:13:28 PM

Task #2 (0.67 pts.) - Related URLs

Progress: 100%

Details:

- Include the direct link to this file from the Milestone branch (should end in .php)
- Include the heroku prod link to this file (Just grab the base prod url and manually enter the path to the file)
- Include the related pull request link for this feature

URL #1 https://github.com/Nate-Gaw/ng569-IT202-450/Milestone1/public_html/project/login.php — https://github.com/Nate-Gaw

URL #2 https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/login.php — https://ng569-it202-450-prod

URL #3 https://github.com/Nate-Gaw/ng569-IT202-450/pull/25 — https://github.com/Nate-Gaw

URL #4 https://github.com/Nate-Gaw/ng569-IT202-450/pull/24 — https://github.com/Nate-Gaw

URL #5 https://github.com/Nate-Gaw/ng569-IT202-450/pull/23 — https://github.com/Nate-Gaw

URL #6 https://github.com/Nate-Gaw/ng569-IT202-450/pull/22 — https://github.com/Nate-Gaw

URL #7 https://github.com/Nate-Gaw/ng569-IT202-450/pull/21 — https://github.com/Nate-Gaw

URL #8
https://github.com/Nate-Gaw/ng569-IT202-450/pull/20

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

URL #9
https://github.com/Nate-Gaw/ng569-IT202-450/pull/19

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

URL #10
https://github.com/Nate-Gaw/ng569-IT202-450/pull/18

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

URL #11
https://github.com/Nate-Gaw/ng569-IT202-450/pull/16

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

Saved: 7/7/2025 5:29:13 PM

Task #3 (0.67 pts.) - User Session Evidence

Progress: 100%

Details:

```
- From the heroku logs (Dashboard -> More button -> view Logs), capt
- It should show the id, username, email, and roles
```

Heroku Logs

Saved: 7/7/2025 5:36:33 PM

Section #3: (1 pt.) Feature: User Will Be Able To Logout

Progress: 100%

Task #1 (1 pt.) - Evidence

Progress: 100%

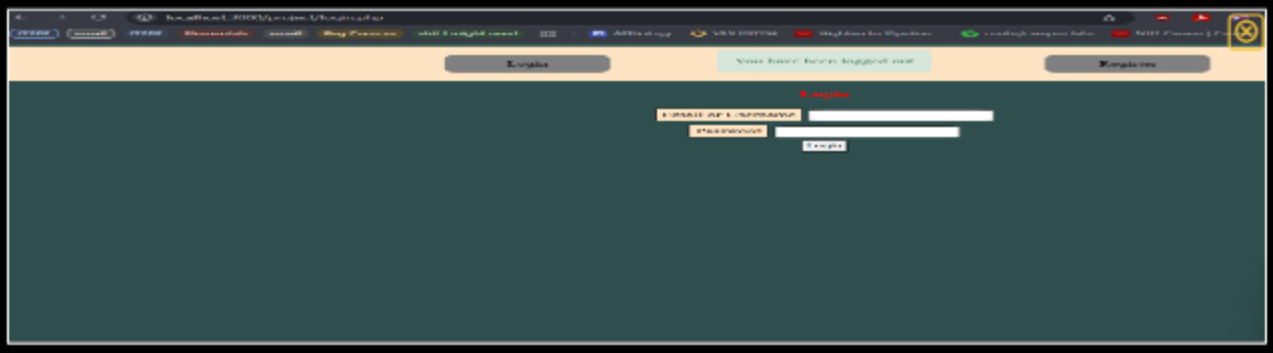
Progress: 100%

Part 1:

Progress: 100%

Details:

- Show the successful logout message
- Show the code snippet of the logout page



Logout Redirect

```php
<?php
// require functions.php to pull in flash()
require(__DIR__ . "/../../lib/functions.php");
reset_session(); // clear session data and start a new session
flash("You have been logged out","success");
header("Location: $BASE_PATH/login.php"); // redirect back to login
```

Logout Code

```php
<?php
function reset_session()        You, 6 days ago • clean up milestone1 .
{
    // ensure session is started before attempting to reset
    // just because it's not active doesn't mean it doesn't exist
    if(session_status() !== PHP_SESSION_ACTIVE){
        session_start();
        error_log("Session was not active, started a new session.");
    }
    session_unset();
    session_destroy();
    session_start();
} <- #3-13 function reset_session()
```

Reset Session Function

Saved: 7/7/2025 2:45:24 AM

Part 2:

Progress: 100%

Details:

- Include the pull request url for this feature

URL #1
https://github.com/Nate-Gaw/ng569-IT2025/7

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

Saved: 7/7/2025 2:45:24 AM

Part 3:

Progress: 100%

Details:

- Briefly explain how the logout process works and how the user is officially logged off

Your Response:

The reset_session function calls session functions, which destroys and starts a new session so the other session is terminated. Then the alert is issues and the user is redirected back to the login page.

Saved: 7/7/2025 2:45:24 AM

Section #4: (2 pts.) Feature: Security Rules

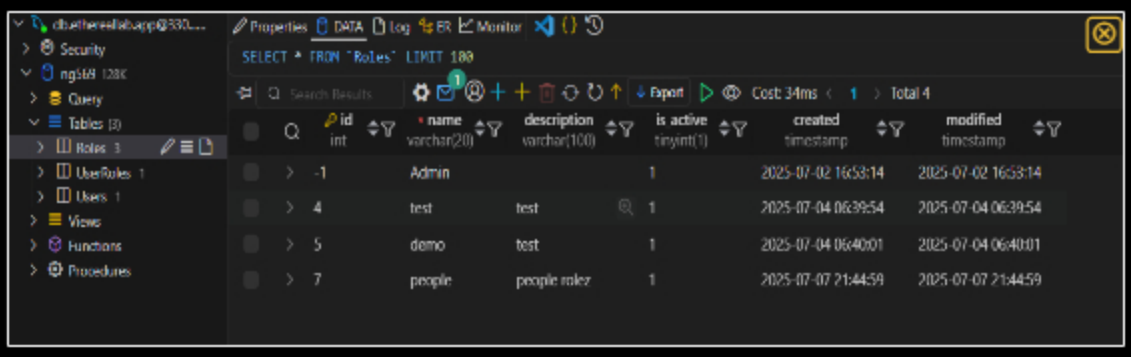
Progress: 100%

Task #1 (1 pt.) - Database Tables

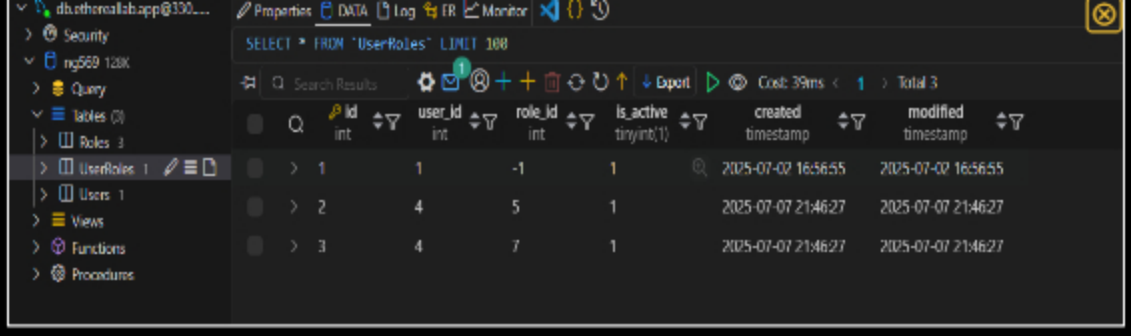
Progress: 100%

Details:

```
- From the vs code mysql tool, show both the Roles and UserRoles tal
```



Roles DB



UserRoles DB

Saved: 7/7/2025 5:47:26 PM

Task #2 (1 pt.) - Demo Security Rules

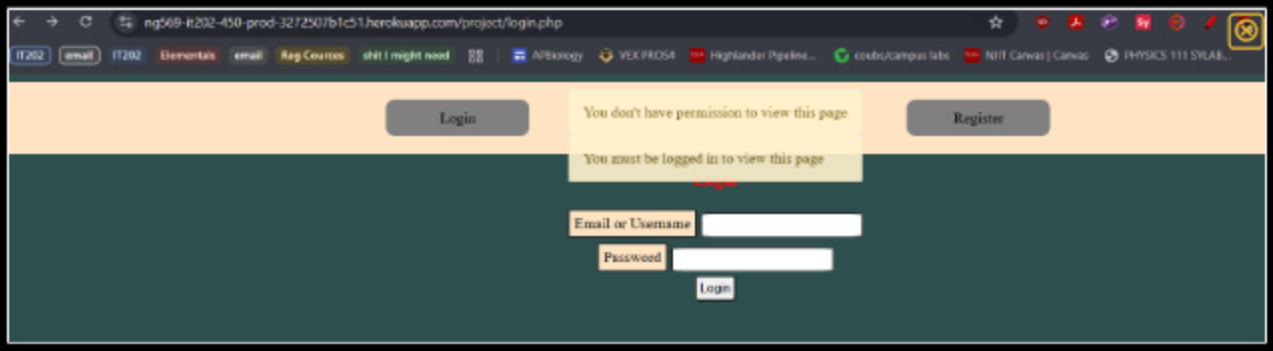
Progress: 100%

Part 1:

Progress: 100%

Details:

- Show the message related to needing to be logged in (i.e., try to manuall
- Show the message related to not having the proper permission/role (i.e.,
- Include a snippet of the login check function
- Include a snippet of the role check function



Shows both message to be needing to logged in to access page and not having proper role

```php
<?php
//ng569 7/7/25 redirect if user is not logged in
require_once(__DIR__ . "/../../partials/nav.php");
if (!is_logged_in()) {
    die(header("Location: login.php"));
}
?>
```

Check if user is logged in (from Profile.php)

Check if user is logged in (from Profile.php)

```php
//ng569 7/7/25 checks to see if user has admin role and is allowed in
if (!has_role("Admin")) {
    flash("You don't have permission to view this page", "warning");
    die(header("Location: " . get_url("landing.php")));
}
//attempt to apply
```

Check if user has role (from assign_roles.php)

Function snippet (from user_helpers.php)

Saved: 7/7/2025 6:03:52 PM

Part 2:

Progress: 100%

Details:

- Include the pull request url for this feature

URL #1
https://github.com/Nate-Gaw/ng569-IT202-450/2
URL
https://github.com/Nate-Gaw/ng5

URL #2
https://github.com/Nate-Gaw/ng569-IT202-450/1
URL
https://github.com/Nate-Gaw/ng5

URL #3
https://github.com/Nate-Gaw/ng569-IT202-450/8
URL
https://github.com/Nate-Gaw/ng5

Saved: 7/7/2025 6:03:52 PM

Part 3:

Progress: 100%

Details:

- Briefly explain how the login check code works
- Briefly explain how the role check code works

Your Response:

Since user data is stored in $_SESSION, the code loops through the array to figure out if the user is still logged in and any roles that the user has. If the code finds nothing, the user is redirected to another page and given an alert. If the code does find something, but it doesn't align with the roles, the same thing happens. After every time logout is clicked, the session is destroyed and erased, so previous information will not get mixed up

Saved: 7/7/2025 6:03:52 PM

Section #5: (2 pts.) Feature: User Profile

Progress: 100%

Task #1 (1 pt.) - Validation

Progress: 100%

Details:

- Show the relevant code snippets for each validation layer
- Show the relevant demo of each validation layer
- Briefly explain how the validation steps work at each layer

Task #1 (0.33 pts.) - HTML Form/Validation

Progress: 100%

Details:

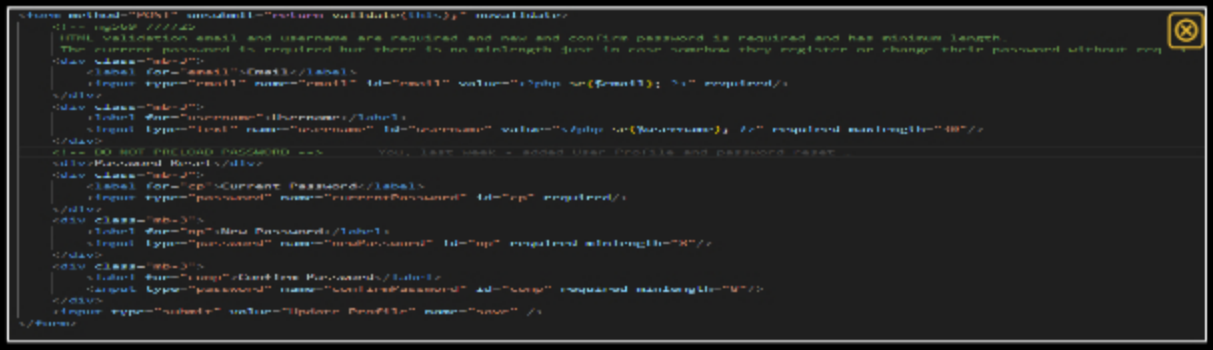
- System should allow the user to correct the error without wiping/clearing the form (for the PHP side this includes sticky-form logic to keep the username/email address entries)

Part 1:

Progress: 100%

Details:

- Show the code related to the profile page's form (HTML validation)
- Show examples of each validation message (you may be able to capture this in one or few screenshots)



HTML Validation code

Heroku HTML Validation

Heroku HTML Validation 2

Saved: 7/7/2025 10:02:21 PM

Part 2:

Progress: 100%

Details:

- Briefly explain the validation step (i.e, what you chose, how they solve the requirement)

Your Response:

The code has everything required, so things are not able to be erased or entered as empty. Passwords are then req and have min length > 8 except the current password, in case of mix up. The php validation will check to make sure the password is correct anyway.

Saved: 7/7/2025 10:02:21 PM

Task #2 (0.33 pts.) - JS Validation (validate() function)

Progress: 100%

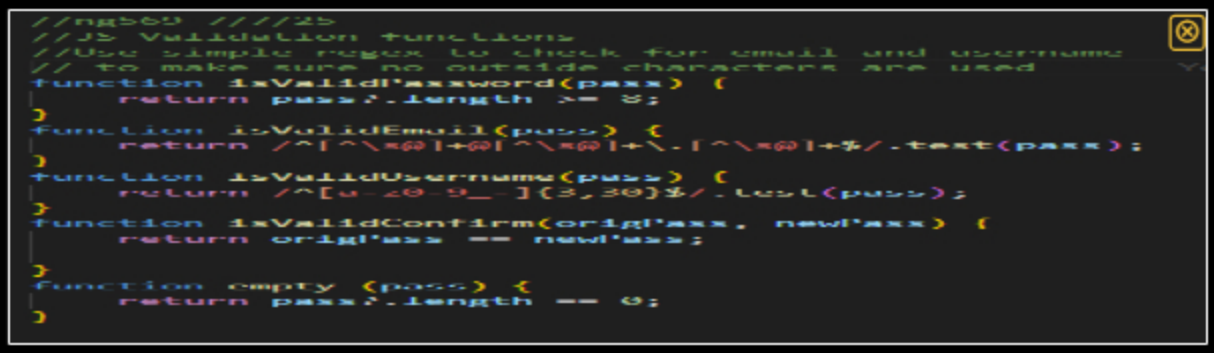
Part 1:

Progress: 100%

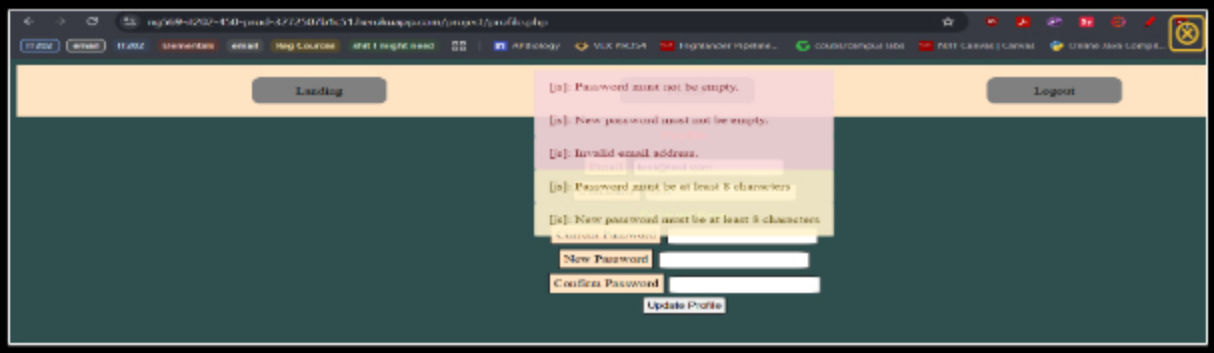
Details:

- Show the code related to the profile page's validate() function
- Show examples of each validation message [username format, email format, password format, password doesn't match confirm] (you may be able to capture this in one or few screenshots)
- Note: You may need to use dev tools to temporarily apply `novalidate` to the form tag

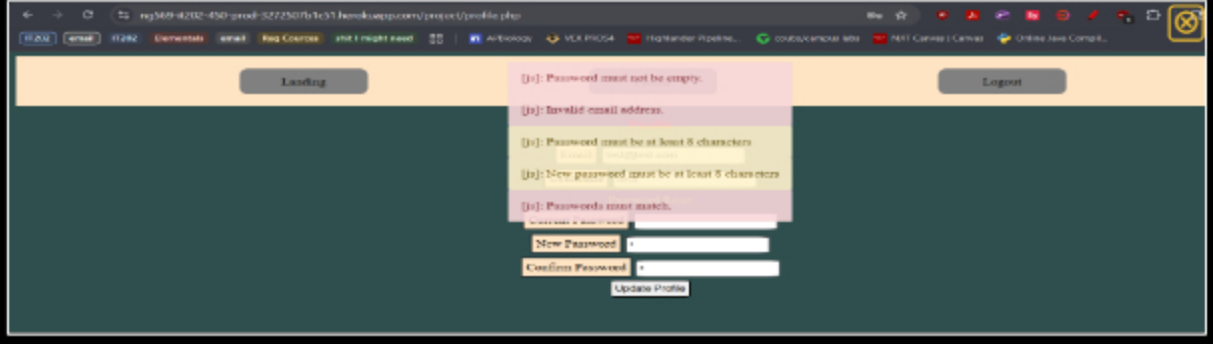
Javascript Code



JS Validation Functions



Heroku Javascript Validations



Heroku Javascript Validations 2

Saved: 7/7/2025 9:07:08 PM

Part 2:

Progress: 100%

Details:

- Briefly explain the validation step (i.e, what you chose, how they solve the requirement)

Your Response:

This code reuses the javascript functions I created, which uses regex and checks length of strings to make sure the passwords, email and username format are valid.

Saved: 7/7/2025 9:07:08 PM

Task #3 (0.33 pts.) - PHP Validation (steps before the DB call)

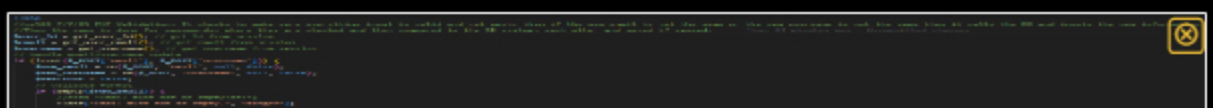
Progress: 100%

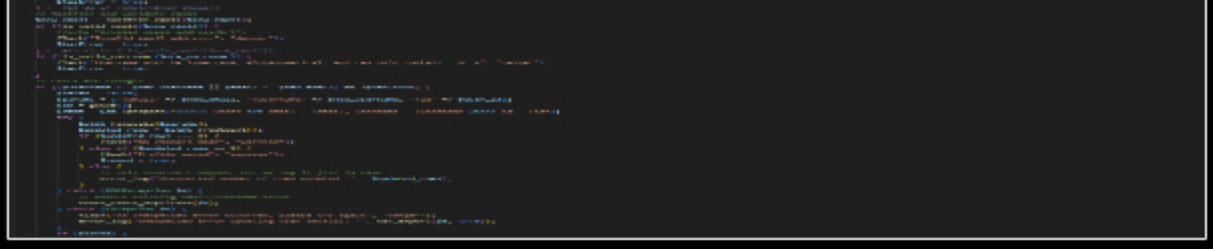
Part 1:

Progress: 100%

Details:

- Show the code related to the profile page's PHP validation
- Show examples of each validation message [username format, email format, password format, invalid password, password doesn't match confirm, username taken, email taken] (you may be able to capture this in one or few screenshots)
- Note: You may need to use dev tools to temporarily apply `novalidate` to the form tag and temporarily override the validate() function or use the provided helper script in the instructions

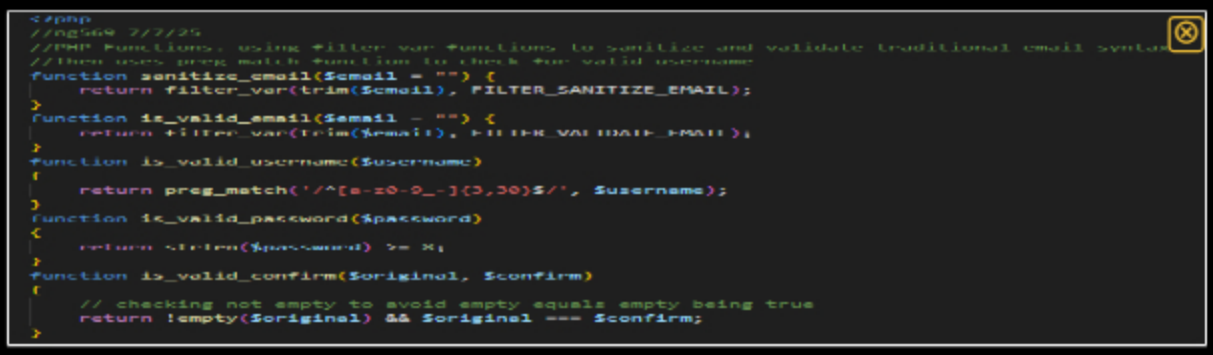




PHP Validation Code

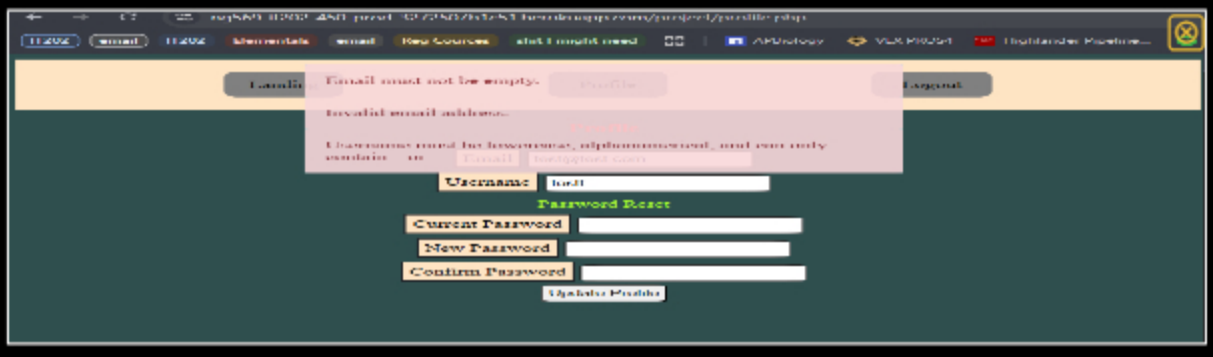
PHP Validation Code 2

PHP Validation Code 3



```php
<?php
//ng568 7/7/25
//PHP Functions, using filter_var functions to sanitize and validate traditional email syntax
//then uses preg_match function to check for valid username
function sanitize_email($email = "") {
    return filter_var(trim($email), FILTER_SANITIZE_EMAIL);
}
function is_valid_email($email = "") {
    return filter_var(trim($email), FILTER_VALIDATE_EMAIL);
}
function is_valid_username($username)
{
    return preg_match('/^[a-z0-9_-]{3,30}$/', $username);
}
function is_valid_password($password)
{
    return strlen($password) >= 8;
}
function is_valid_confirm($original, $confirm)
{
    // checking not empty to avoid empty equals empty being true
    return !empty($original) && $original === $confirm;
}
```

PHP Validation Functions



Heroku PHP

Heroku PHP 2

Heroku PHP 3

Saved: 7/7/2025 9:33:51 PM

Part 2:

Progress: 100%

Details:

- Briefly explain the validation step (i.e, what you chose, how they solve the requirement)

Your Response:

The php validation checks for the same thing JS does, except it actually validates it with the DB. First it checks for the email and username and adjusts them if they are valid. Then the current password is checked with the DB and then the new and confirm are cross-checked, assuming they are in a valid format. Then the new password is saved in the DB.

Saved: 7/7/2025 9:33:51 PM

Task #2 (1 pt.) - Related URLs

Progress: 100%

Details:

- Include the direct link to this file from the Milestone branch (should end in .php)
- Include the heroku prod link to this file (Just grab the base prod url and manually enter the path to the file)
- Include the related pull request link for this feature

URL #1
https://github.com/Nate-Gaw/ng569-IT202-450/Milestone1/public_html/project/profile.php
URL: https://github.com/Nate-Gaw

URL #2
https://ng569-it202-450-prod-3272507b1c51.herokuapp.com/project/profile.php
URL: https://ng569-it202-450-prod

URL #3
https://github.com/Nate-Gaw/ng569-IT202-450/pull/24
URL: https://github.com/Nate-Gaw

URL #4
https://github.com/Nate-Gaw/ng569-IT202-450/pull/21
URL: https://github.com/Nate-Gaw

URL #5
https://github.com/Nate-Gaw/ng569-IT202-450/pull/20
URL: https://github.com/Nate-Gaw

URL #6
https://github.com/Nate-Gaw/ng569-IT202-450/pull/25
URL: https://github.com/Nate-Gaw

Saved: 7/7/2025 9:36:42 PM

Section #6: (1 pt.) Misc

Progress: 100%

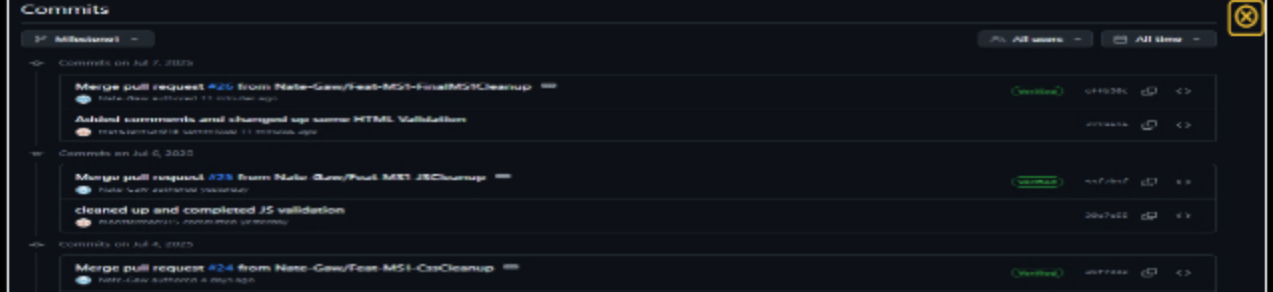
Task #1 (0.33 pts.) - Github Details

Progress: 100%

Part 1:

Progress: 100%

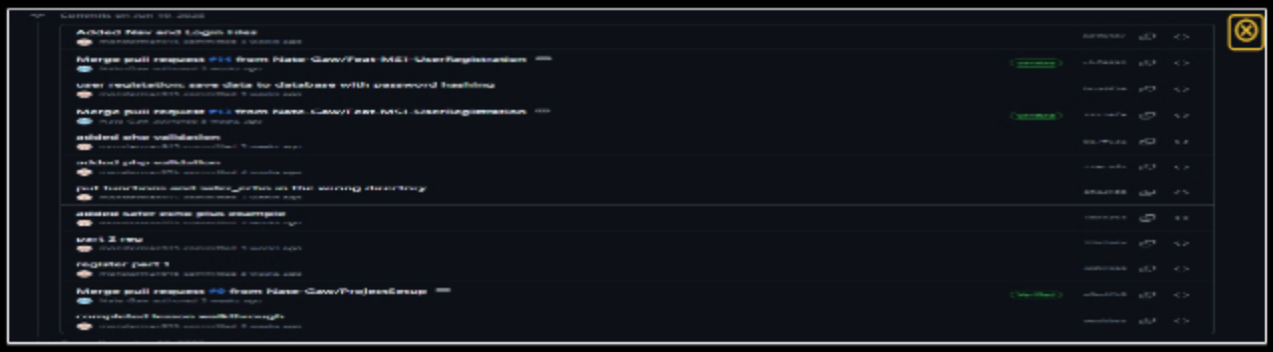
Details:
From the Commits tab of the Pull Request screenshot the commit history



Commit History

Commit History 2

Commit History 3



Commit History 4

Saved: 7/7/2025 10:02:58 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-IT202-M5026

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

Saved: 7/7/2025 10:02:58 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

Wakatime branch and file

Saved: 7/7/2025 10:05:36 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 web development. I never realized how much actually went to just having a user create an account and login. I always thought it was a simple click type and save. But I learned that there is a lot more running in the background to make sure that your account is safe and secure. I also learned how to program in PHP, javascript and HTML. I had done a bit of everything in my high school years, but nothing to this extent. The same goes for all of my CS classes at NJIT. Like sure, some are hard to understand, but this has been the coolest and most practical thing I have done. Furthermore, I learned about how Sessions and DB works, which I will need to know and understand for my next semester, so it's been really cool learning about all of it and getting a head start is always great.

Saved: 7/7/2025 10:08:45 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:

The easiest part has to be the javascript. Since it was all based on the PHP validations, it went by pretty fast and smoothly. Of course I still learned how to use regex and some functions in javascript, but since the structure was all there, it was a lot easier than understanding the PHP for sure.

Saved: 7/7/2025 10:09:55 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 has to be understanding the PHP portion. Since the class is online, it's been hard to understand how PHP works at times. I had to

even watch a 3 hr course on SQL to understand how it all came together. But I can say it is very neat how these things work and the things you can do with PHP. But nonetheless, looking at the massive amount of code I have, I don't think I would've been able to write this within the summer by myself and especially considering I don't know much about PHP. Furthermore, this time limit made it kinda hard to sit down and really look into the PHP side of things. A lot of it I can see and understand, Im just not sure how I would've made it from scratch, which is kinda the hard part of any assignment/project. This is a really cool project in my opinion, though and I definitely see myself using this knowledge in the real world (for one of the first times in any class)

Saved: 7/7/2025 10:13:57 PM