Course: IT202-008-S2025
Assignment: IT202 Milestone 1
Student: Dylan W. (dw347)
Status: Submitted | Worksheet Progress: 88.46%
Potential Grade: 7.70/10.00 (77.00%)
Received Grade: 0.00/10.00 (0.00%)
Grading Link: https://learn.ethereallab.app/assignment/v3/IT202-008-S2025/it202-milestone-1/grading/dw347

Instructions

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

Task #1 (0.67 pts.) - Validation

Weight: *33.33%*
Objective: *Validation*

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

Sub-Tasks:

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

Combo Task:

Weight: *33.33%*
Objective: *HTML Form/Validation*

Details:

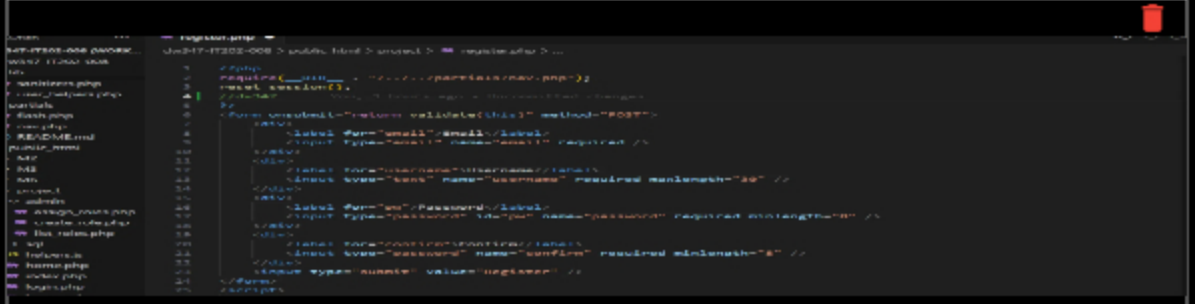
- System should allow the user to correct the error without wiping/clearing the form (for the PHP side this includes sticky-form logic to keep the username/email address entries)

Image Prompt

Weight: *50%*

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)



code

pw too short

email taken

email wrong

passwords dont match

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

Combo Task:

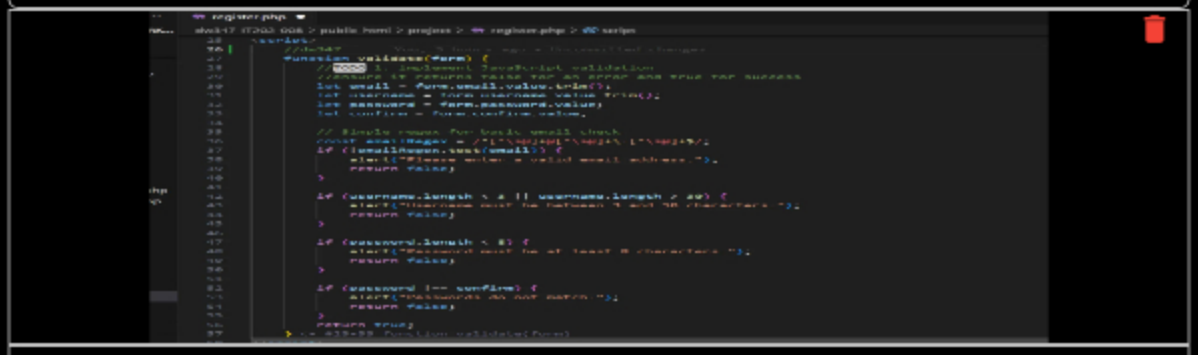
Weight: *33.33%*
Objective: *JS Validation (validate() function)*

Image Prompt

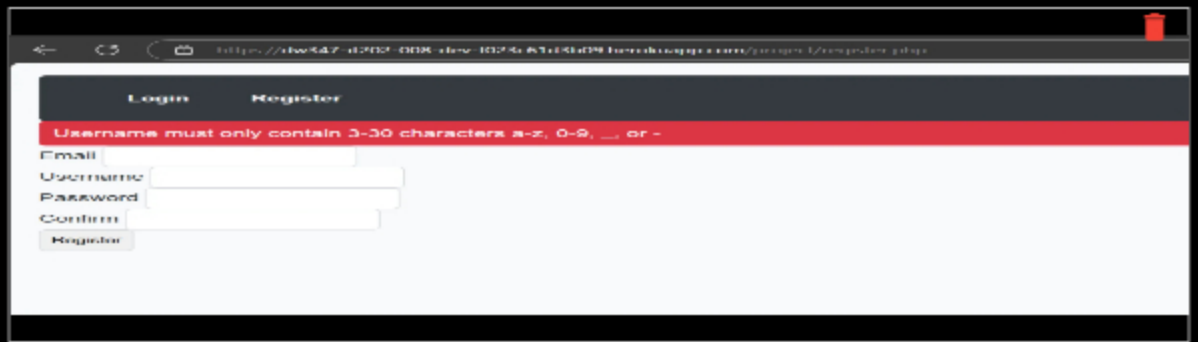
Weight: *50%*

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



code



user wrong

email wrong

user too long or short

password too short

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

Combo Task:

Weight: *33.33%*

Objective: *PHP Validation (steps before the DB call)*

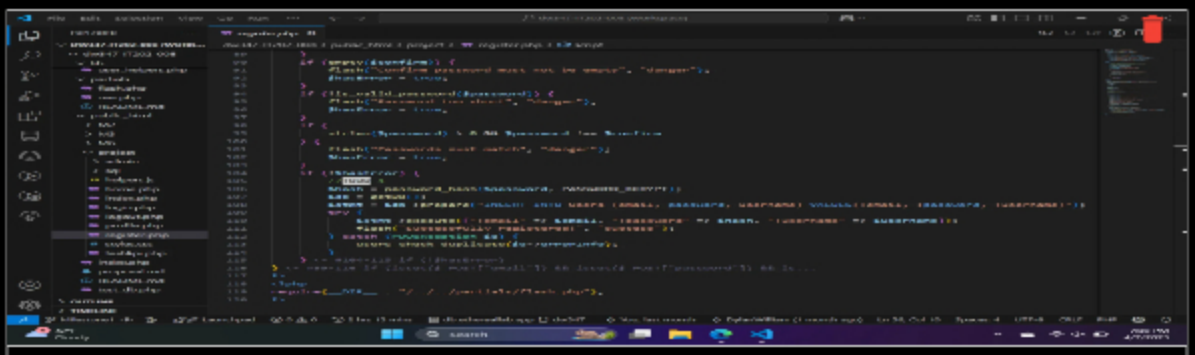
Image Prompt

Weight: *50%*

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

code

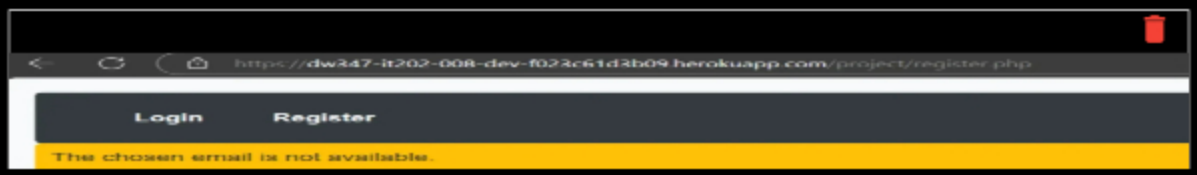


code

email cant be empty

user cant be empty

password dont match



Task #2 (0.67 pts.) - Related URLs

Url Prompt

Weight: *33.33%*
Objective: *Related URLs*

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/dyylan2018/dw347-IT202-008/Milestone1/public_html/project/register.php

URL #2

https://dw347-it202-008-prod-39957dda50b0.herokuapp.com/project/register.php

URL #3

https://github.com/dyylan2018/dw347-IT202-008/blob/Milestone1/public_html/project/register.php

Saved: 4/7/2025 7:23:38 PM

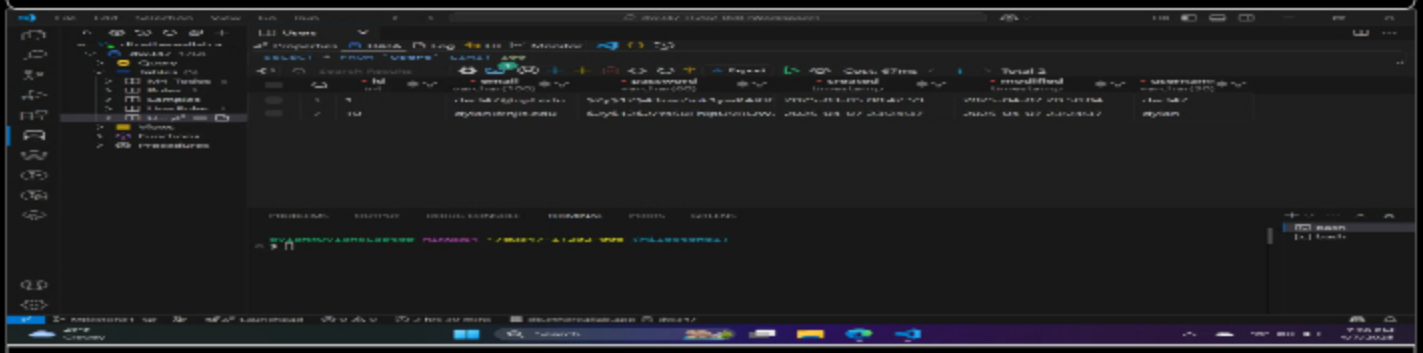
Task #3 (0.67 pts.) - Database - Users table

Image Prompt

Weight: *33.33%*
Objective: *Database - Users table*

Details:

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



users table

Saved: 4/7/2025 7:26:31 PM

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

Task #1 (0.67 pts.) - Validation

Weight: *33.33%*
Objective: *Validation*

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

Sub-Tasks:

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

Combo Task:

Weight: *33.33%*
Objective: *HTML Form/Validation*

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)

Image Prompt

Weight: *50%*

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)

code

sticky form keeps user entry while PW is wrong

Saved: 4/7/2025 9:52:20 PM

Text Prompt

Weight: *50%*

Details:

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

Your Response:

This login form combines client-side and server-side validation for security and usability: it uses JavaScript (`validate()`) for preliminary checks on submission (though not shown here, likely verifying email/username format and password requirements), while the `novalidate` attribute overrides default HTML5 validation to allow custom checks. The form submits via POST to securely transmit credentials, and the email input is protected against XSS attacks using `htmlspecialchars()` to escape previously entered values. HTML5 attributes (`required`, `minlength="8"`) enforce non-empty fields and an 8-character

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

Combo Task:

Weight: *33.33%*

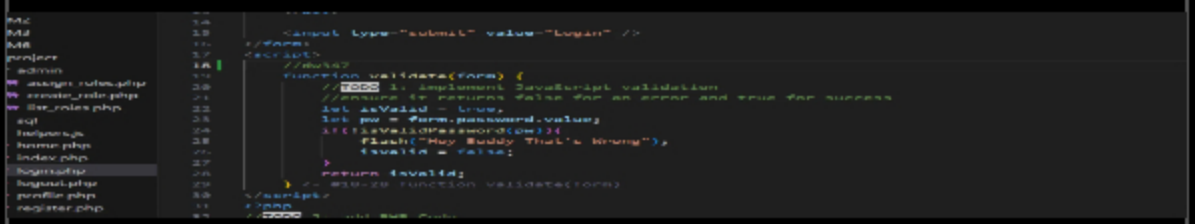
Objective: *JS Validation (validate() function)*

Image Prompt

Weight: *50%*

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



code

password too short

Saved: 4/7/2025 9:53:01 PM

Text Prompt

Weight: *50%*

Details:

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

Your Response:

This JavaScript validation function checks if the password meets security requirements by calling `isValidPassword()` (a presumed helper function that verifies password strength), displaying an error message via `flash()` if validation fails, and returning `false` to prevent form submission; if valid, it returns `true`, allowing the form to proceed, ensuring only properly formatted passwords are submitted while providing user feedback.

Saved: 4/7/2025 9:53:01 PM

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

Combo Task:

Weight: *33.33%*
Objective: *PHP Validation (steps before the DB call)*

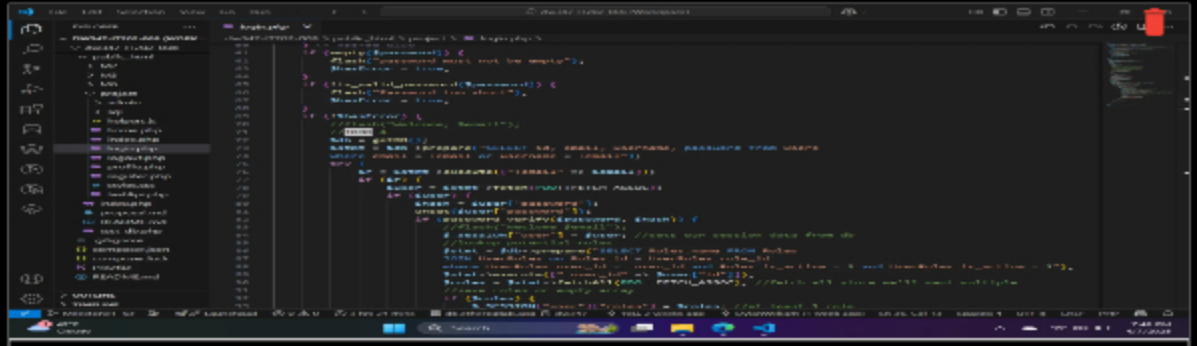
Image Prompt

Weight: *50%*

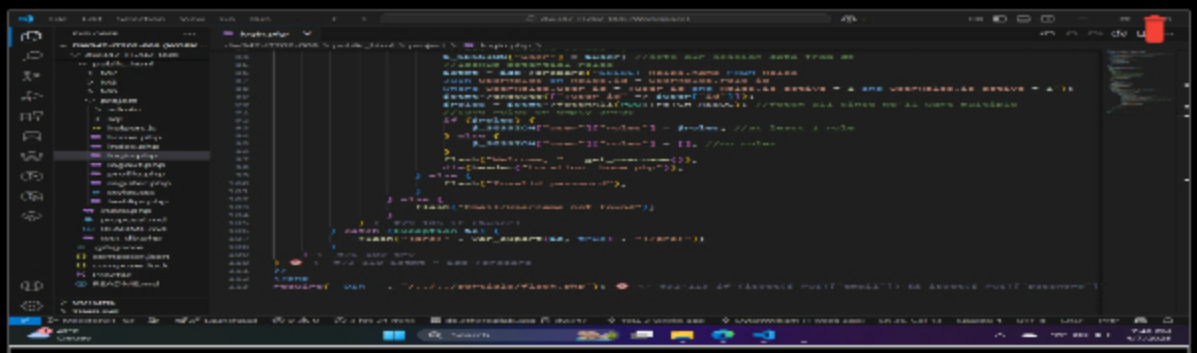
Details:

- 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

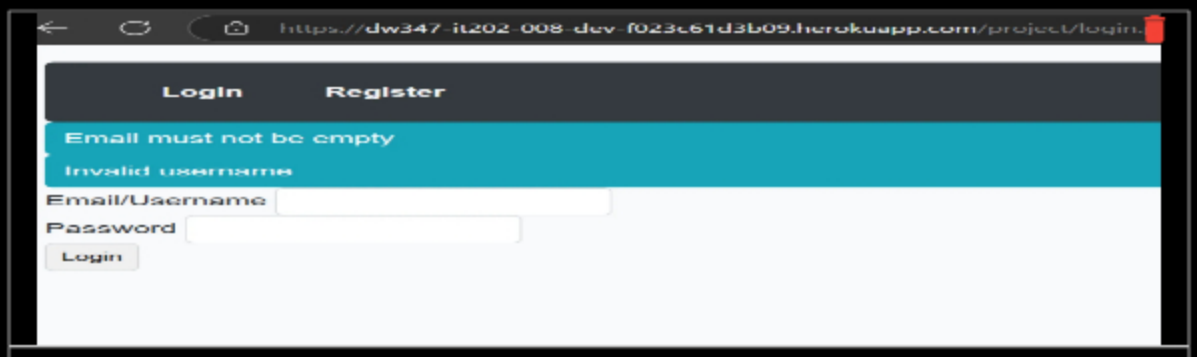
code



code

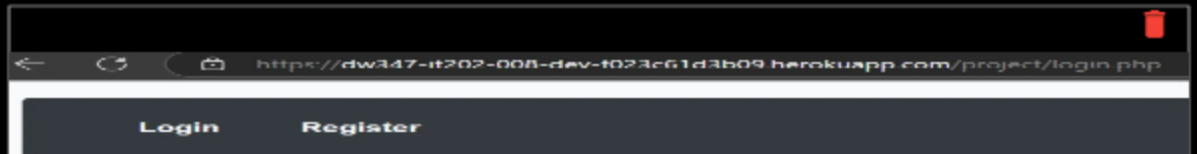


code



invalid user and email empty

login successful



Hey Buddy That's Wrong

Task #2 (0.67 pts.) - Related URLs

Url Prompt

Weight: *33.33%*
Objective: *Related URLs*

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/dyylan2018/dw347-IT202-008/Milestone1/public_html/project/login.php

URL
https://github.com/dyylan2018/dw347

URL #2
https://dw347-it202-008-prod-39957dda50b0.herokuapp.com/project/login.php

URL
https://dw347-it202-008-prod-39957dd

URL #3
https://github.com/dyylan2018/dw347-IT202-0069

URL
https://github.com/dyylan2018/dw347

Saved: 4/7/2025 8:04:24 PM

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

Image Prompt

Weight: *33.33%*
Objective: *User Session Evidence*

Details:

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

heroku logs

Saved: 4/7/2025 8:11:55 PM

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

Task #1 (1 pt.) - Evidence

Combo Task:

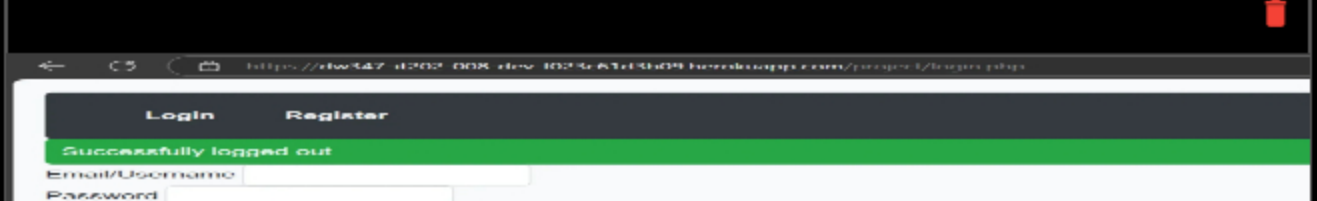
Weight: *100%*
Objective: *Evidence*

Image Prompt

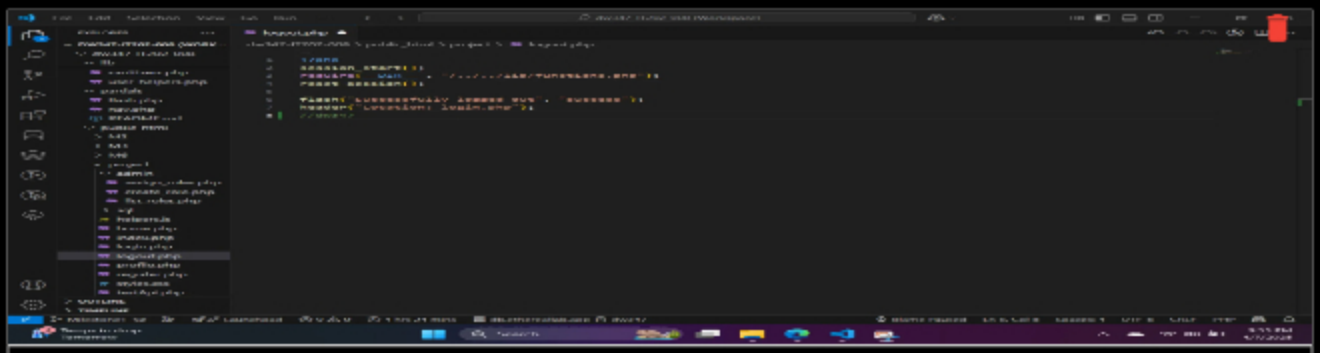
Weight: *40%*

Details:

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



output message



code snippet

Saved: 4/7/2025 9:55:36 PM

Url Prompt

Weight: *20%*

Details:

- Include the pull request url for this feature

URL #1
https://github.com/dyylan2018/dw347-IT202-[illegible]

URL
https://github.com/dyylan2018/dw:

Saved: 4/7/2025 9:55:36 PM

Text Prompt

Weight: *40%*

Details:

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

Your Response:

The logout process securely terminates the user session by first destroying all session data with `session_destroy()`, invalidating the authentication state, then clearing the session cookie by setting its expiration to a past timestamp, ensuring no residual credentials remain on the client side.

and finally redirecting the user to the login page via `header("Location: login.php")` to complete the logout cycle and prevent further access to protected resources.

Saved: 4/7/2025 9:55:36 PM

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

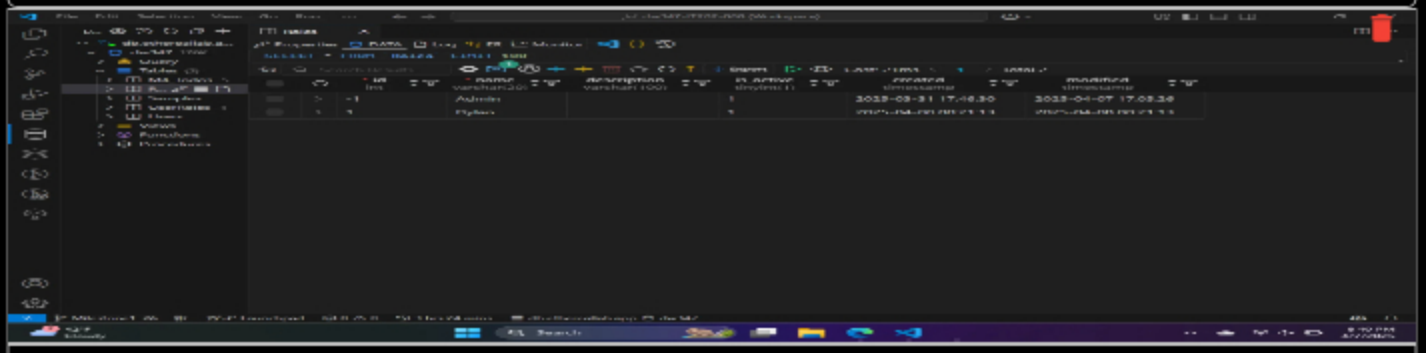
Task #1 (1 pt.) - Database Tables

Image Prompt

Weight: *50%*
Objective: *Database Tables*

Details:

```
- From the vs code mysql tool, show both the Roles and UserRoles tables with data in them
```



roles

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

Combo Task:

Weight: *50%*
Objective: *Demo Security Rules*

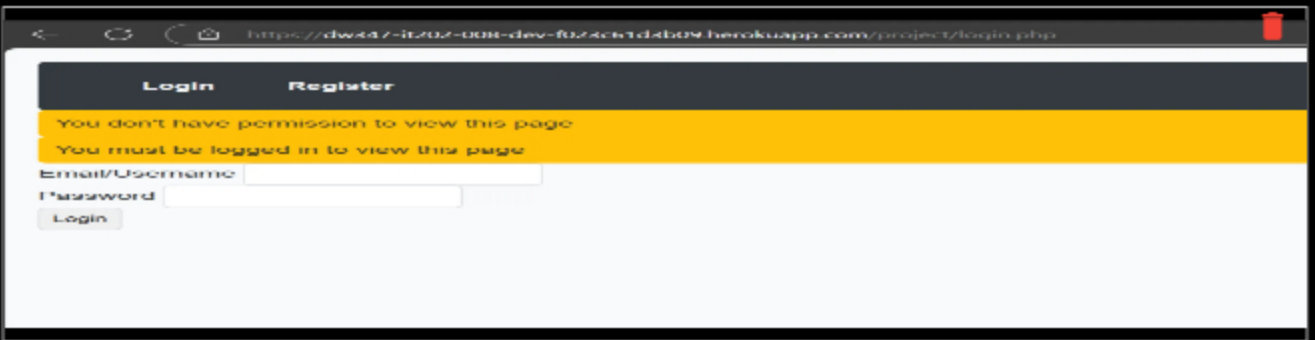
Image Prompt

Weight: *40%*

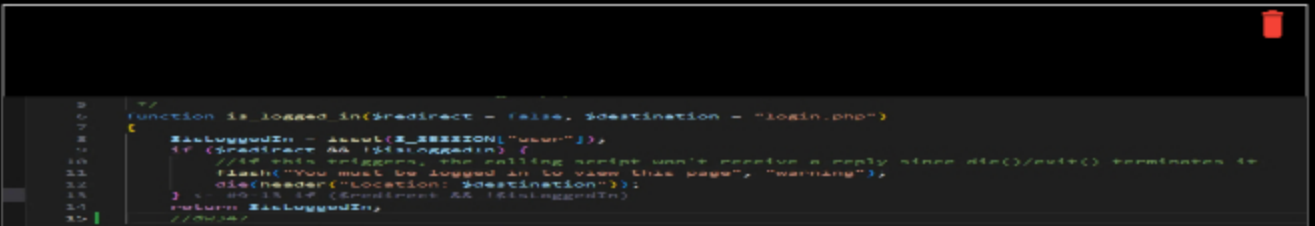
Details:

```
  - Show the message related to needing to be logged in (i.e., try to manually
access a login protected page while logged out)
  - Show the message related to not having the proper permission/role (i.e., try
to manually access a role protected page while not having the proper role)
  - Include a snippet of the login check function
  - Include a snippet of the role check function
```

must be logged in



must have access



login check function

role check function

Saved: 4/7/2025 9:58:19 PM

Url Prompt

Weight: *20%*

Details:

- Include the pull request url for this feature

URL #1
https://github.com/dyylan2018/dw347-
IT20[illegible]

URL
https://github.com/dyylan2018/dw:

Saved: 4/7/2025 9:58:19 PM

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

Task #1 (1 pt.) - Validation

Weight: *50%*
Objective: *Validation*

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

Sub-Tasks:

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

Combo Task:

Weight: *33.33%*
Objective: *HTML Form/Validation*

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)

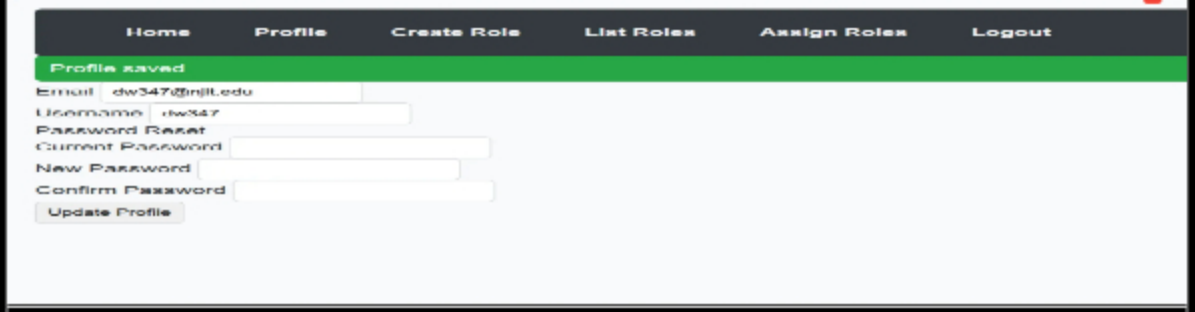
Image Prompt

Weight: *50%*

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)

code snippet



profile saved output

Saved: 4/7/2025 9:13:44 PM

Text Prompt

Weight: *50%*

Details:

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

Your Response:

This form implements secure profile updates with client-side validation (via the `validate()` function) and server-side protection. It collects the user's email and username (pre-filled safely with `se()` to prevent XSS), while requiring the current password for verification and new password fields (not preloaded) for resetting. The form ensures proper password confirmation and uses POST method to securely transmit sensitive data, while the `onsubmit` validation prevents submission if requirements aren't met, balancing usability with security against unauthorized changes.

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

Combo Task:

Weight: *33.33%*
Objective: *JS Validation (validate() function)*

Image Prompt

Weight: *50%*

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

code

PW must match output

Saved: 4/7/2025 9:23:36 PM

Text Prompt

Weight: *50%*

Details:

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

Your Response:

This JavaScript validation function ensures password consistency during profile updates by comparing the new password (`newPassword`) and confirmation (`confirmPassword`) fields, returning `false` and triggering a warning flash message if they don't match (preventing form submission), while maintaining security by performing initial client-side checks before server-side validation processes the sensitive data. The `isValid` flag allows for expandable

server-side validation processes the sensitive data. The `isValid` flag allows for expandable validation (implied by the TODO comment) to incorporate additional checks like password complexity or current password verification in future implementations.

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

Combo Task:

Weight: *33.33%*
Objective: *PHP Validation (steps before the DB call)*

Image Prompt

Weight: *50%*

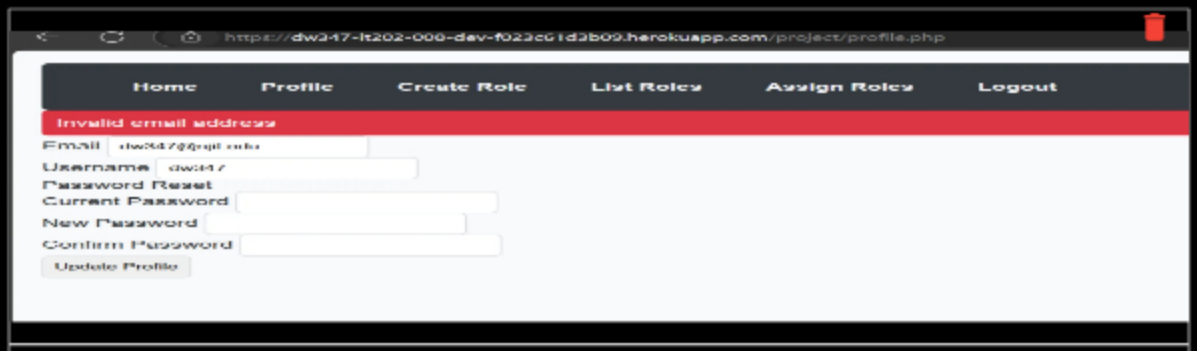
Details:

- 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

code

code

code



wrong email

Task #2 (1 pt.) - Related URLs

Url Prompt

Weight: *50%*
Objective: *Related URLs*

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/dyylan2018/dw347-IT202-008/Milestone1/public_html/project/profile.php

URL
https://github.com/dyylan2018/dw347

URL #2
https://dw347-it202-008-

URL
https://dw347-it202-008-prod-39957dd

prod-39957dda50b0.herokuapp.com/project/profile.php

Section #6: (1 pt.) Misc

Task #1 (0.33 pts.) - Github Details

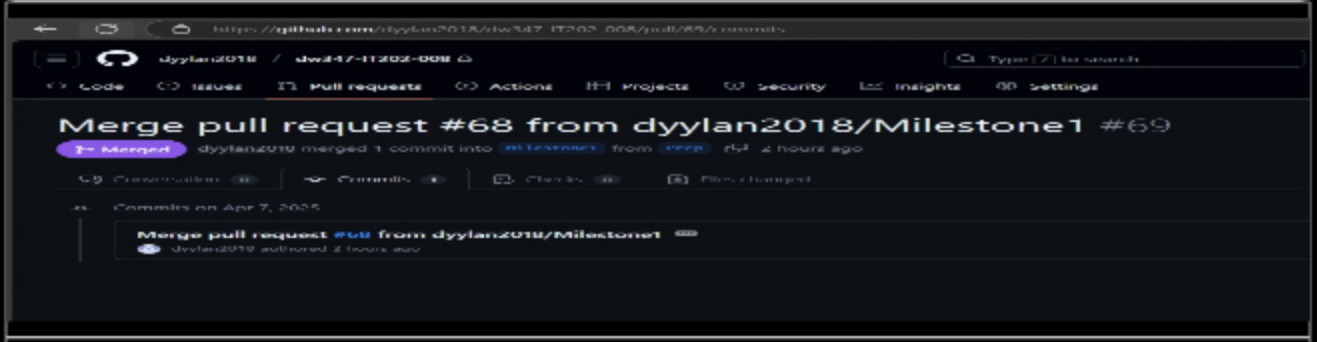
Combo Task:

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

Image Prompt

Weight: *60%*

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



commits

Saved: 4/7/2025 9:42:35 PM

Url Prompt

Weight: *40%*

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

URL #1
https://github.com/dyylan2018/dw347-IT202-[illegible]

URL
https://github.com/dyylan2018/dw:

Saved: 4/7/2025 9:42:35 PM

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

Image Prompt

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

Details:

- Visit the WakaTime.com Dashboard
- Click `Projects` and find your repository
- Capture the overall time at the top that includes the repository name
- Capture the individual time at the bottom that includes the file time
- Note: The duration isn't relevant for the grade and the visual graphs aren't necessary

overall

Task #3 (0.33 pts.) - Reflection

Weight: *33.33%*

Objective: *Reflection*

Sub-Tasks:

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

Text Prompt

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

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

Your Response:

I learned alot about php, javascript, html scripting and how to implement all of that into a webpage.

Saved: 4/7/2025 9:44:34 PM

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

Text Prompt

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

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

Your Response:

the html scripting was the easiest for me

Saved: 4/7/2025 9:45:11 PM

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

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

Text Prompt

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

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

Your Response:

submitting was the hardest (sorry)

Saved: 4/7/2025 9:45:27 PM