Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-milestone-1-2024/grade/tlj3

IT202-008-S2024 - [IT202] Milestone 1 2024

Submissions:

Submission Selection

1 Submission [active] 4/1/2024 6:43:32 PM

Instructions

COLLAPSE

Prereqs:

Go through each lesson from "Project Setup and SQL" to "User Login Enhancement" and follow the branching names while gathering the code
    Merge each into Milestone1 branch
    Mark the related GitHub Issues items as "done"
Implement your own custom CSS (something much different than the default "ugly" CSS given as an example)
    Consider styling all forms/inputs, data output, navigation, etc
Implement JavaScript validation on Register, Logout, and Profile (include "[Client]" in the output messages to differentiate between server-side validations)

Instructions:

Make sure you're in Milestone1 with the latest changes pulled
Ensure Milestone1 has been deployed to heroku dev
Gather the requested evidence and fill in the explanations per each prompt
Save the submission and generate the output PDF
Put the output PDF into your local repository folder
add/commit/push it to GitHub
Merge Milestone1 into dev
Locally checkout dev and pull the changes
Create and merge a pull request from dev to prod to deploy Milestone1 to prod
Upload this output PDF to Canvas

Branch name: Milestone1

Tasks: 26 Points: 10.00

User Registration (2 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Screenshot of form on website page

Checklist

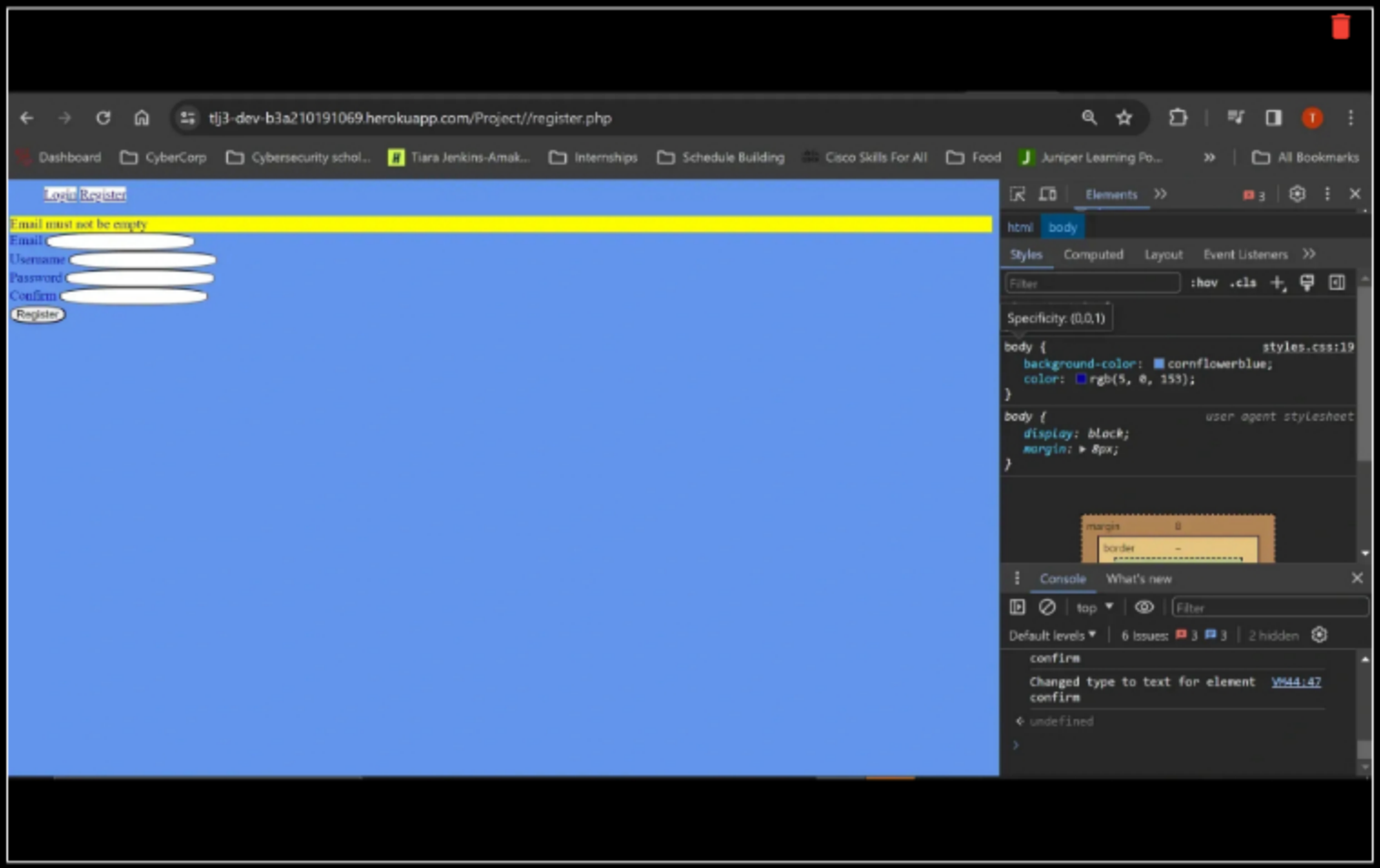
*The checkboxes are for your own tracking

#	Points	Details
#1	1	Heroku dev url should be present in the address bar
#2	1	Should have thoughtful CSS applied
#3	1	Demonstrate JavaScript validation for each field [can be combined] (email validation (format), username validation (format), password validation (format), password and confirm password matching, and each field being required
#4	1	Demonstrate email already in use message (message text doesn't need to be exact, but should be clear)
#5	1	Demonstrate username already in use message (message text doesn't need to be exact, but should be clear)
#6	1	Demonstrate user-friendly message of new account being created

Task Screenshots:

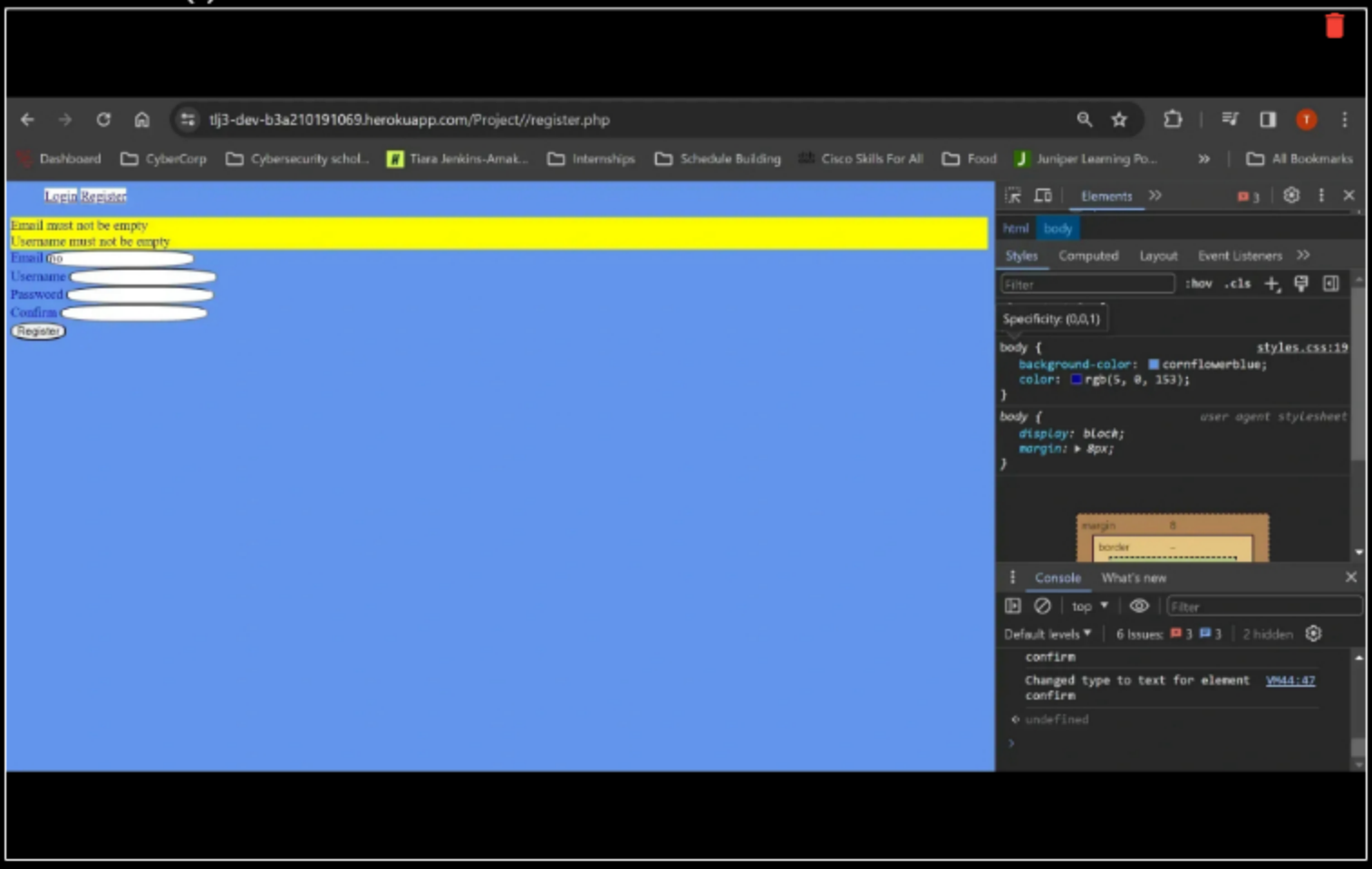
Gallery Style: Large View

Small Medium Large



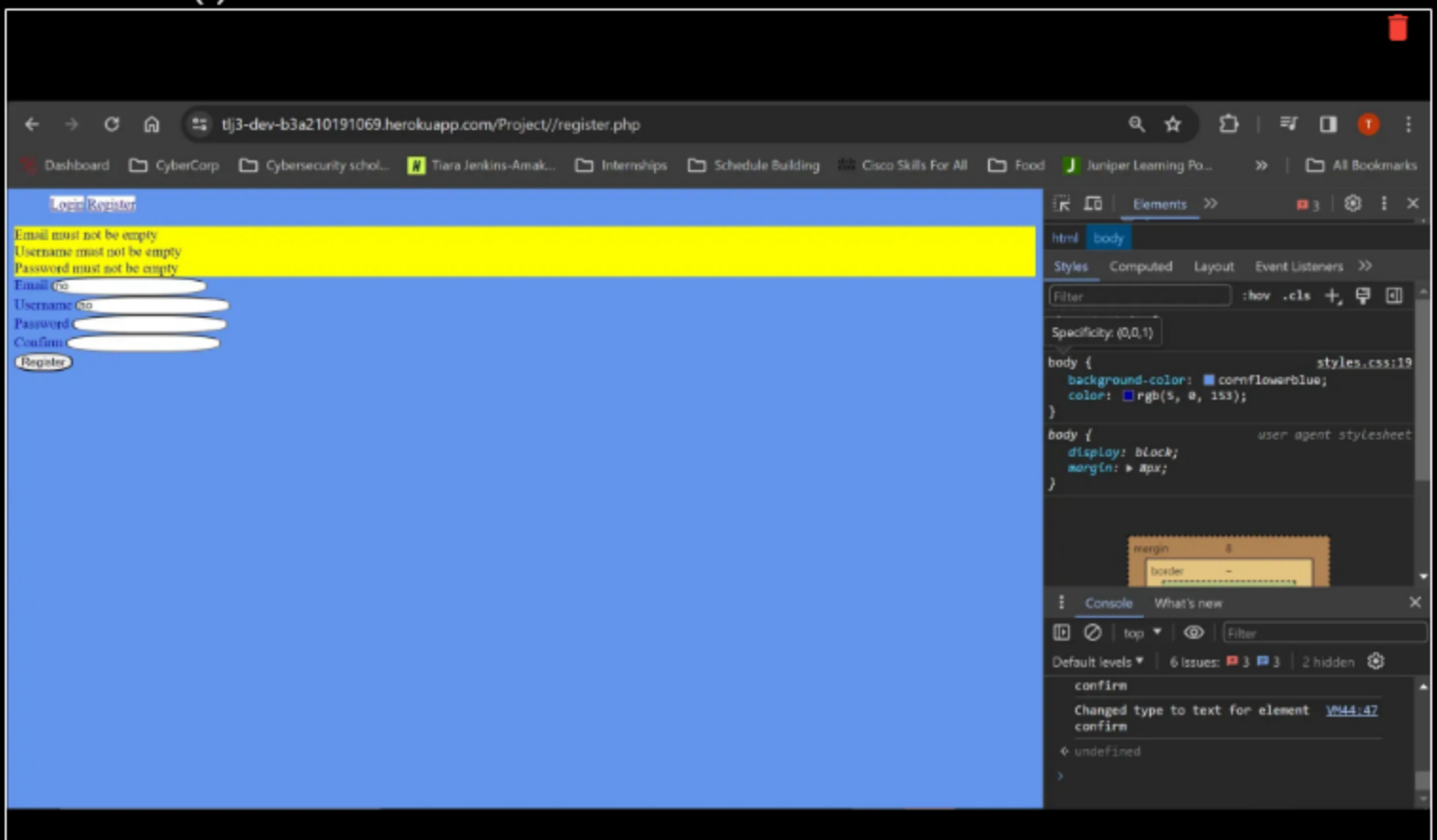
Screenshot shows email is empty

Checklist Items (0)



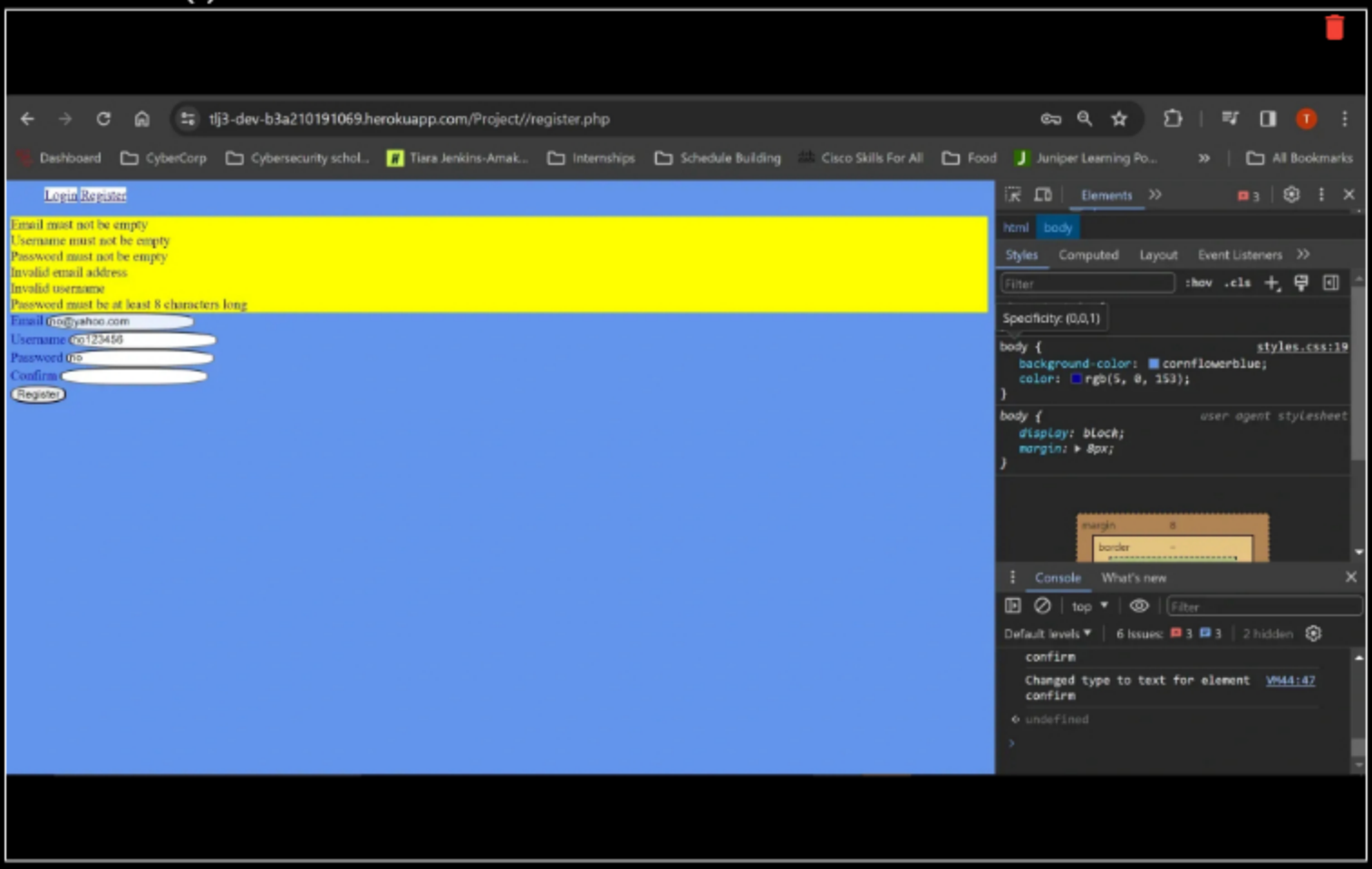
Screenshot shows username is empty

Checklist Items (0)



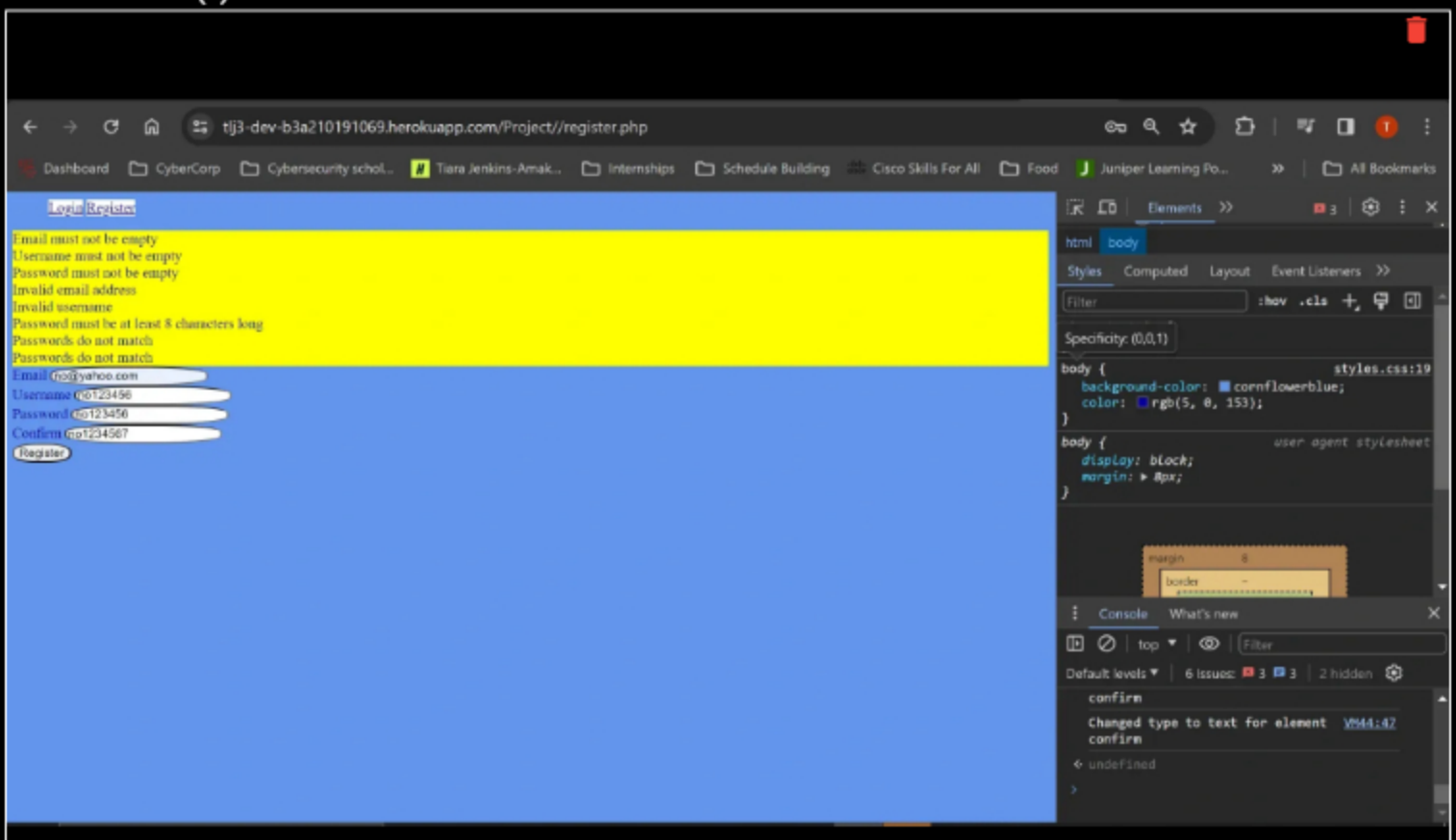
Screenshot shows password is empty

Checklist Items (0)



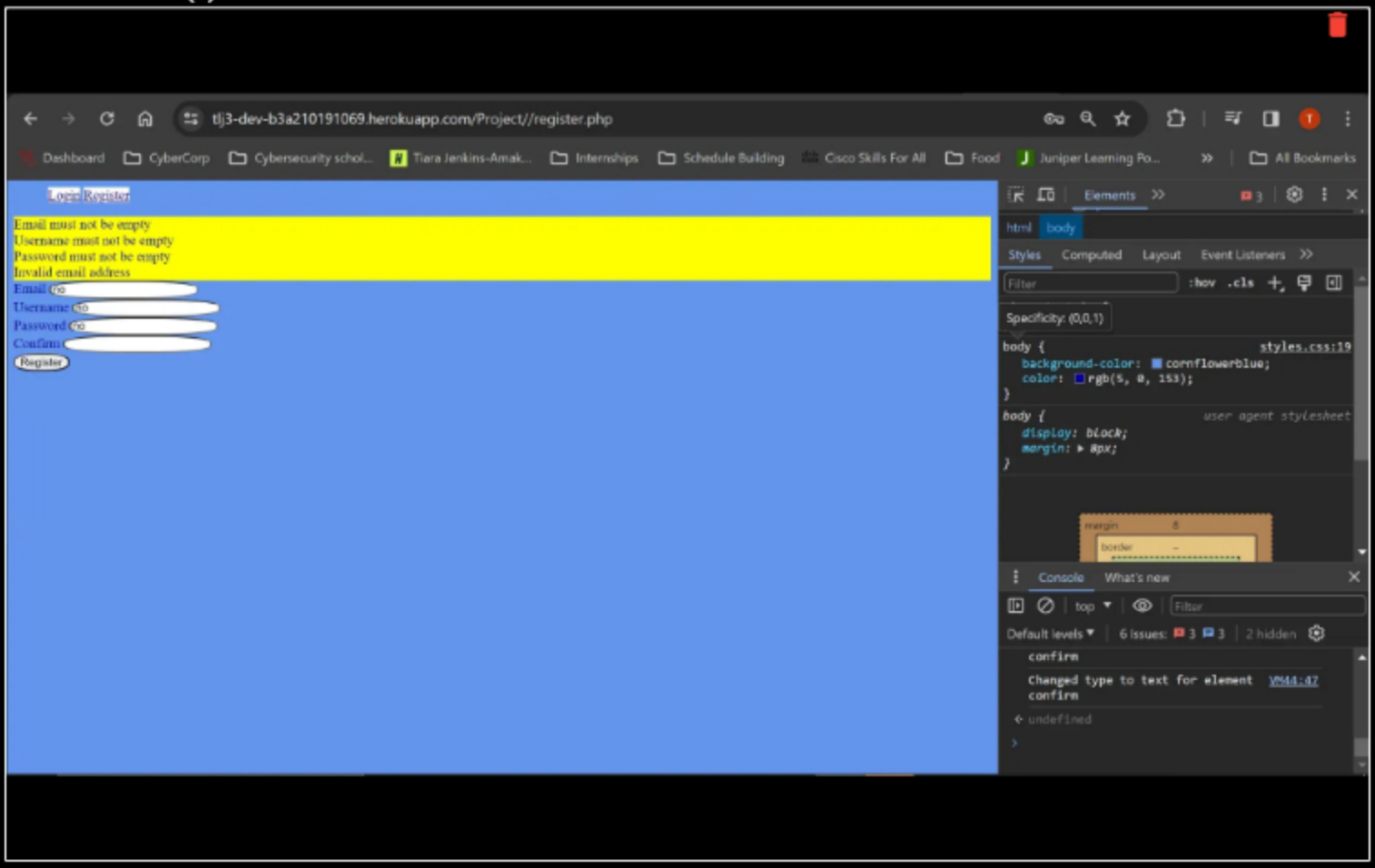
Screenshot shows password needs to be longer

Checklist Items (0)



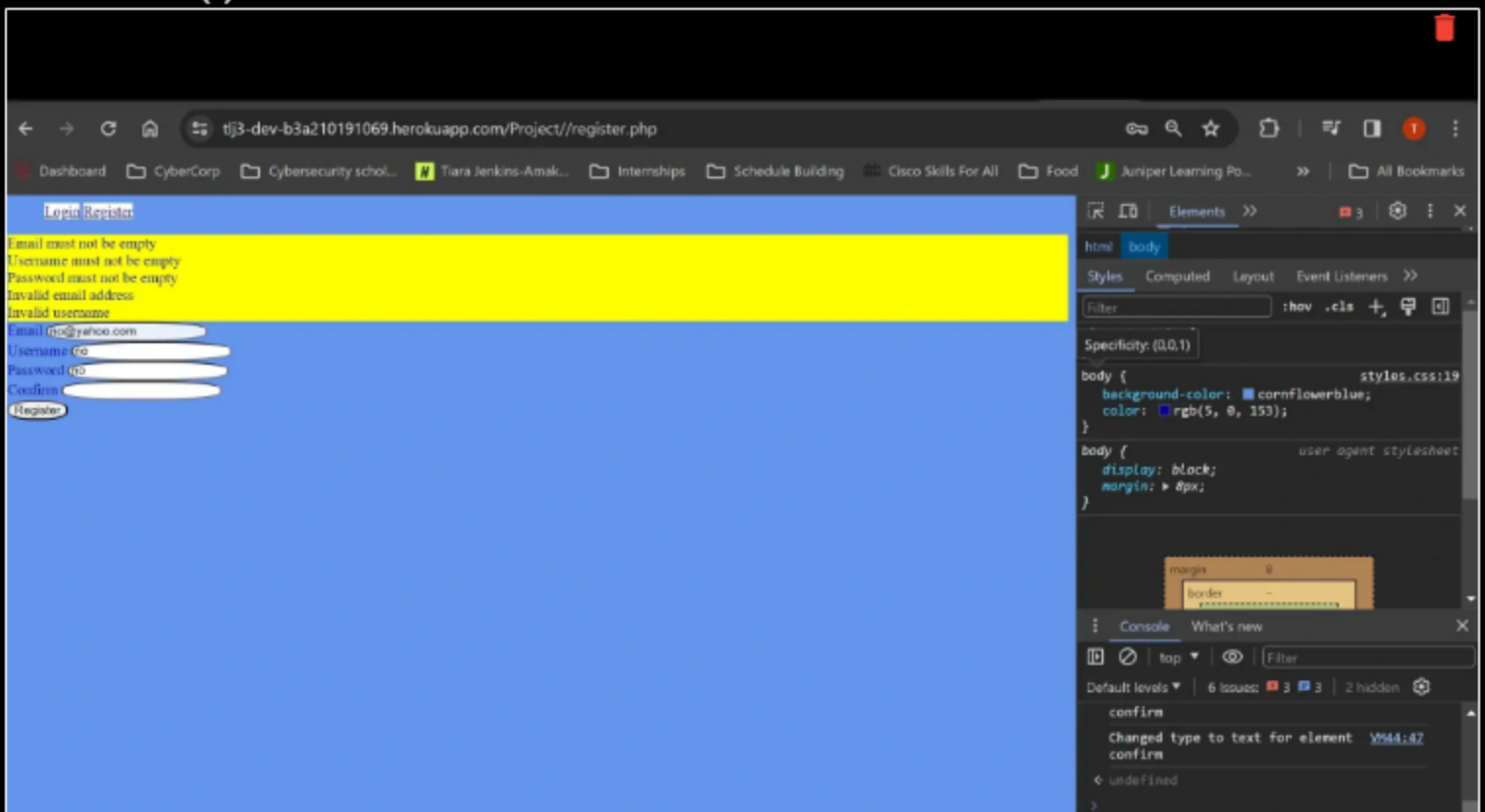
Screenshot shows that the password doesn't match confirm password

Checklist Items (0)



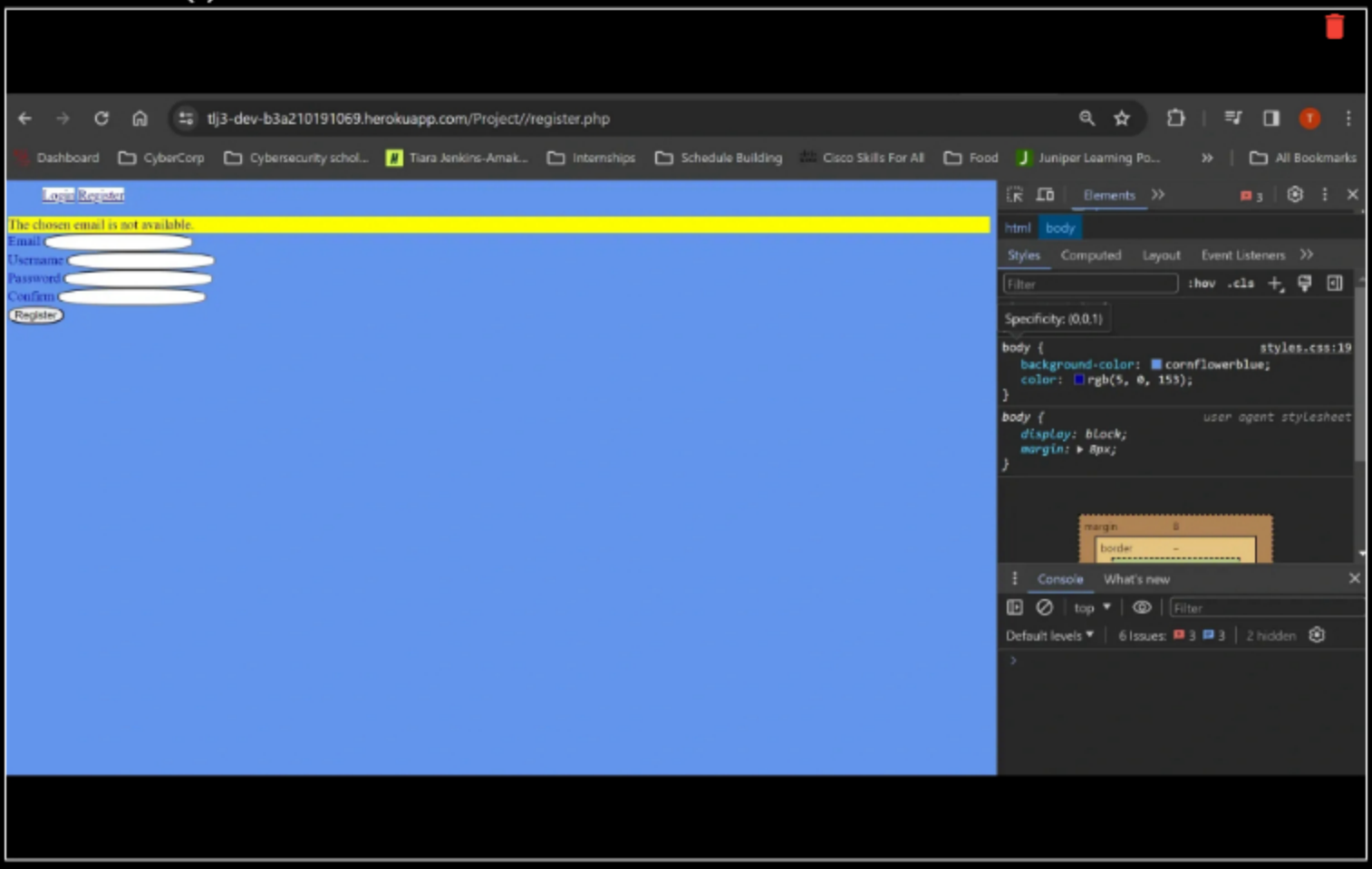
Screenshot shows invalid email

Checklist Items (0)



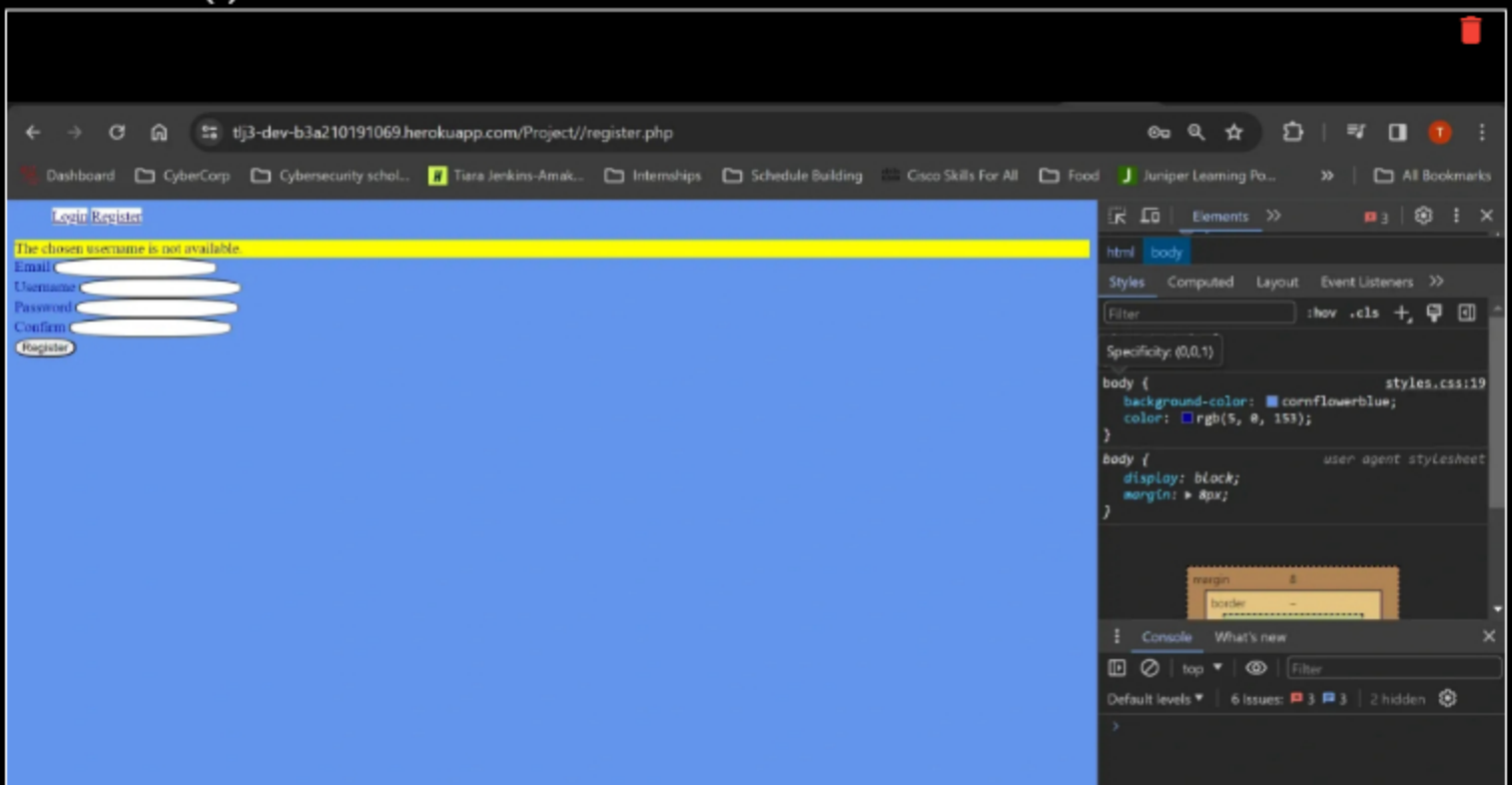
Screenshot shows invalid username

Checklist Items (0)



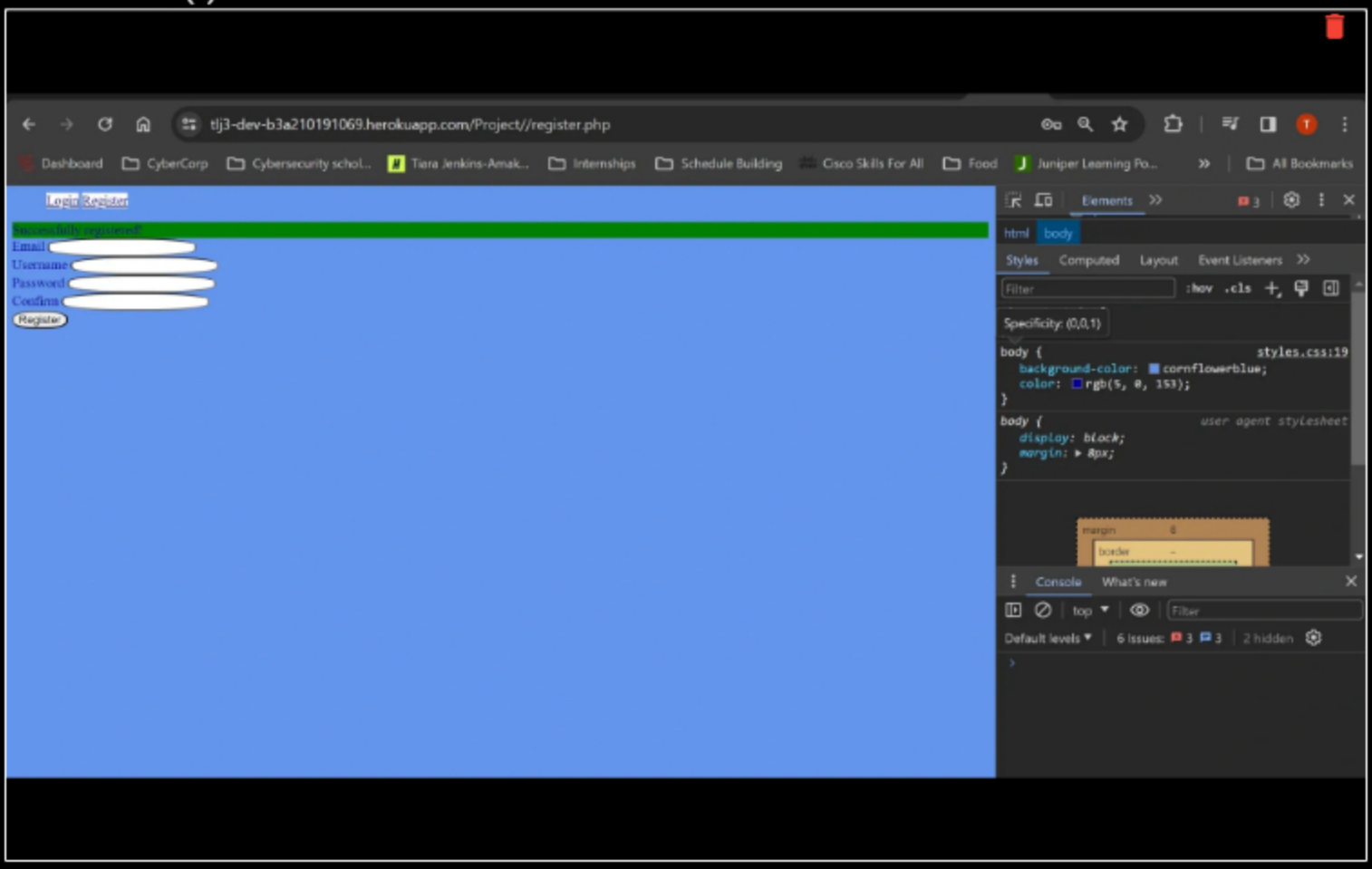
Screenshot shows that the email chosen isn't available

Checklist Items (0)



Screenshot shows that the chosen username isn't available

Checklist Items (0)



Screenshot shows a successful registration

Checklist Items (0)

COLLAPSE

Task #2 - Points: 1

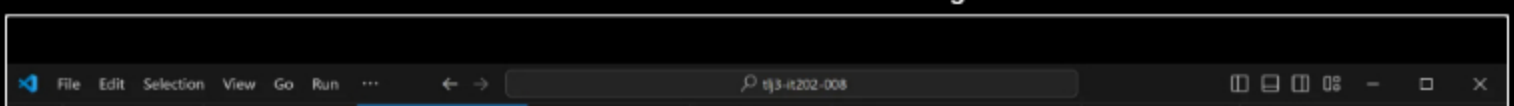
Text: Screenshot of the form code

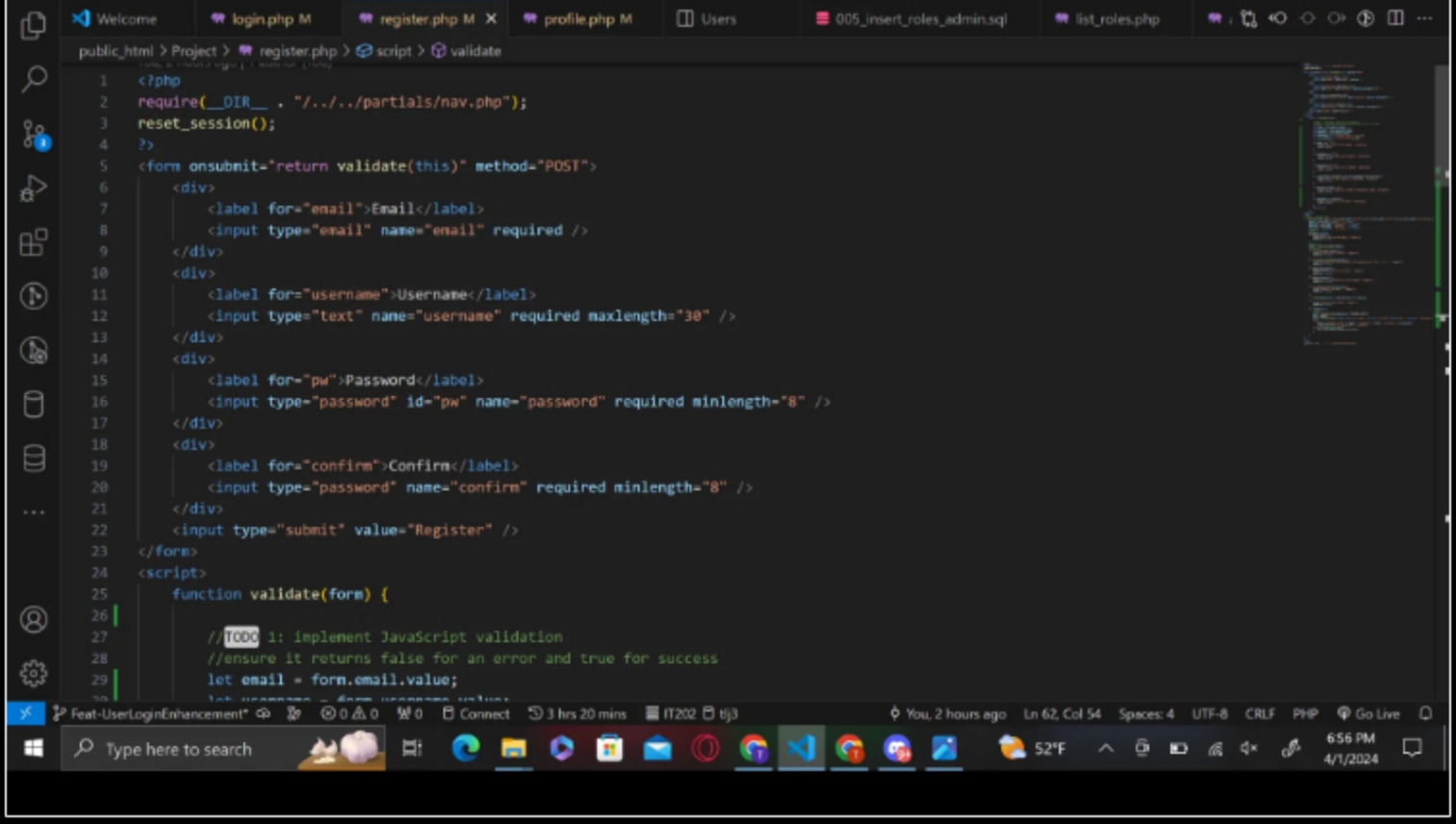
Details:
Should have appropriate input types for the fiel

Task Screenshots:

Gallery Style: Large View

Small Medium Large





Screenshot includes the form code.

COLLAPSE

Task #3 - Points: 1

Text: Screenshot of the client-side and server-side validation code

Checklist

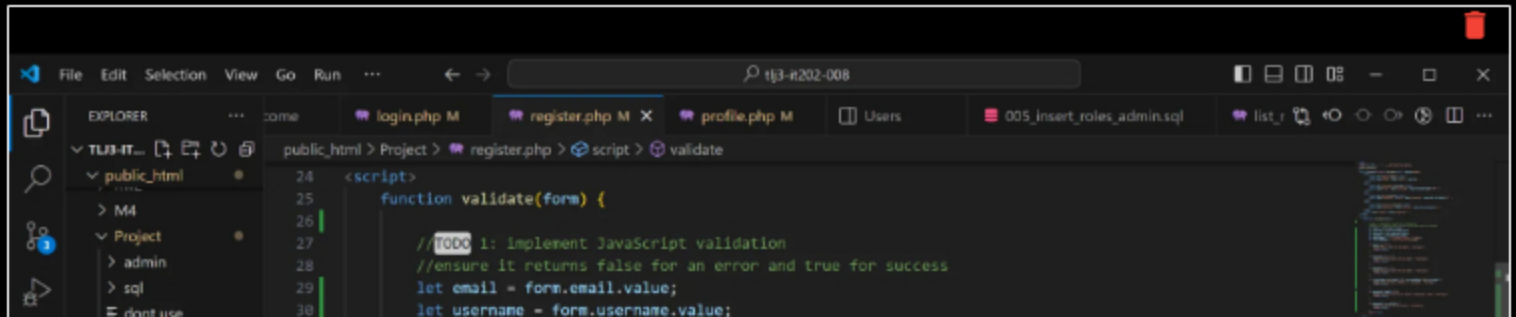
*The checkboxes are for your own tracking

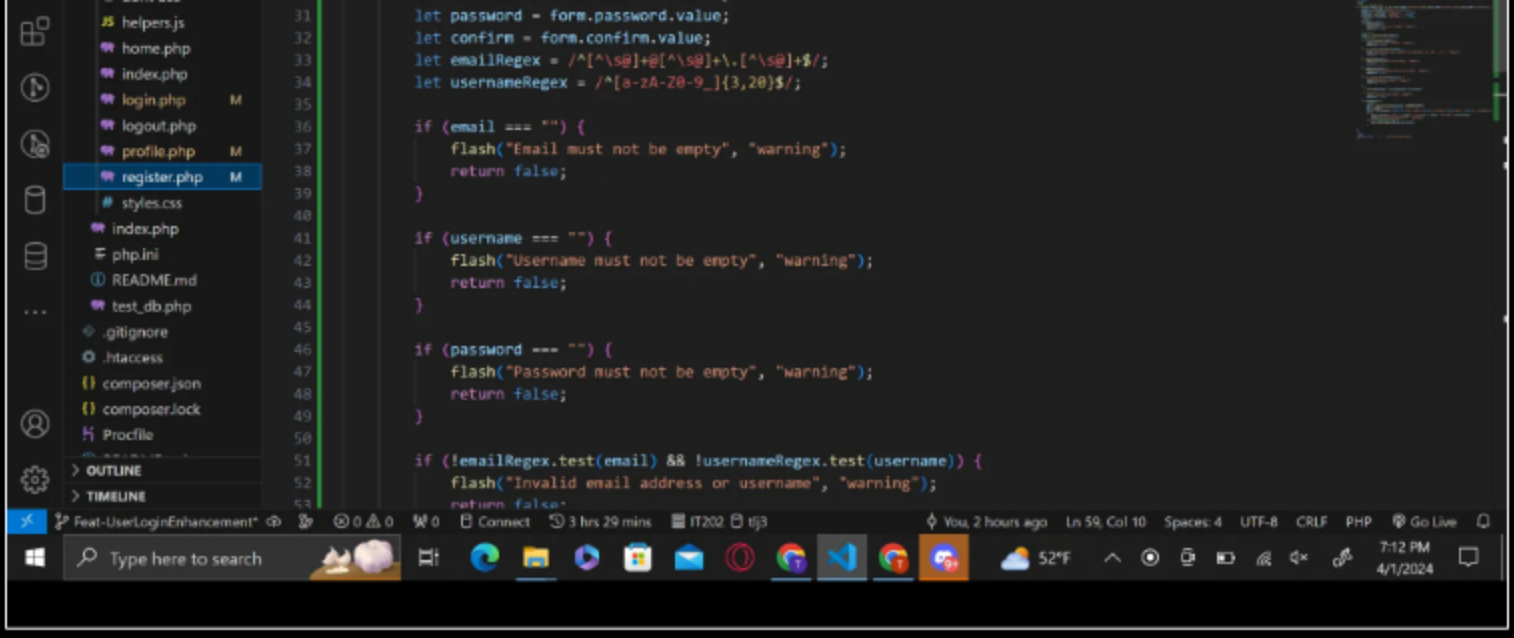
#	Points	Details
#1	1	Show the JavaScript validations (include any extra files related)
#2	1	Show the PHP validations (include any lib content)
#3	1	Include ucid/date code comments in all screenshots (one per screenshot is sufficient)

Task Screenshots:

Gallery Style: Large View

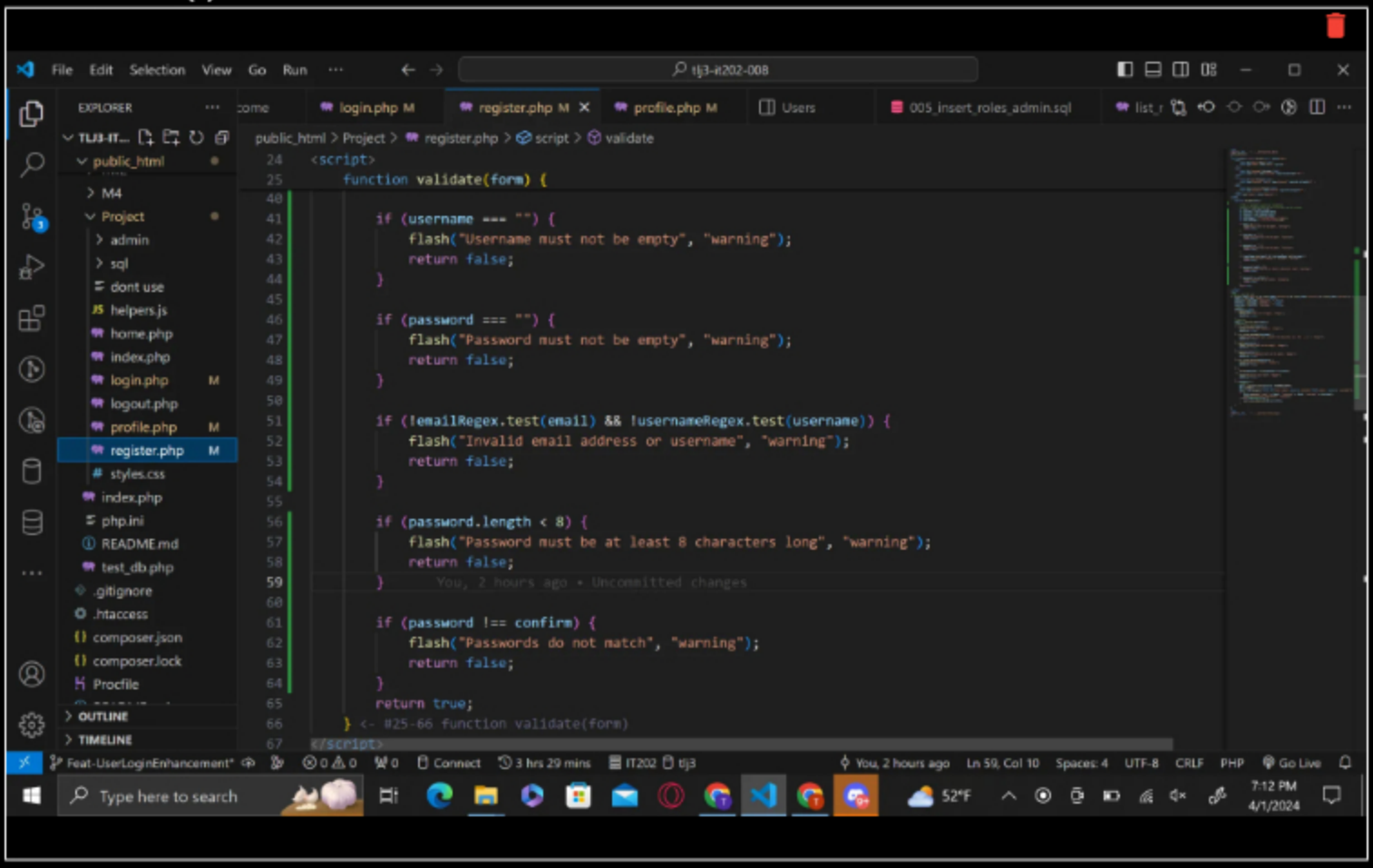
Small Medium Large





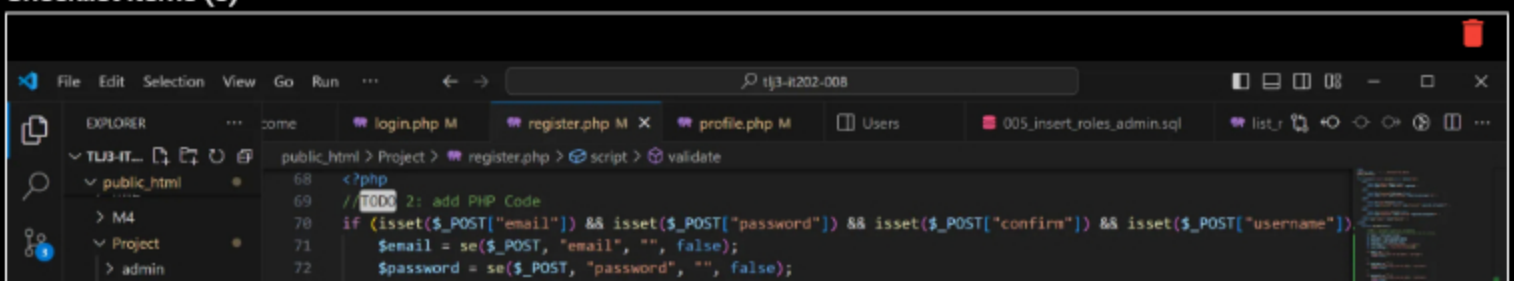
Part 1 of client side validation code.

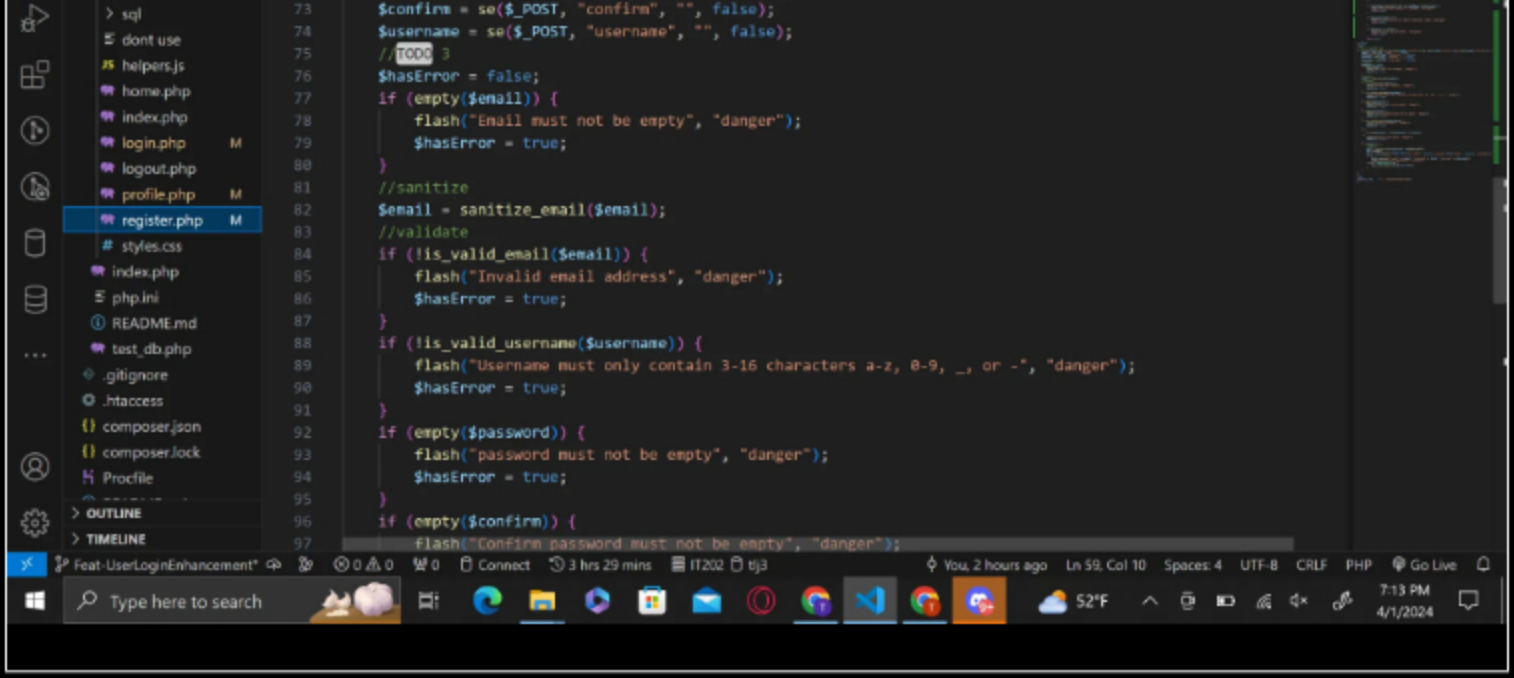
Checklist Items (0)



Part 2 Client Side Validation

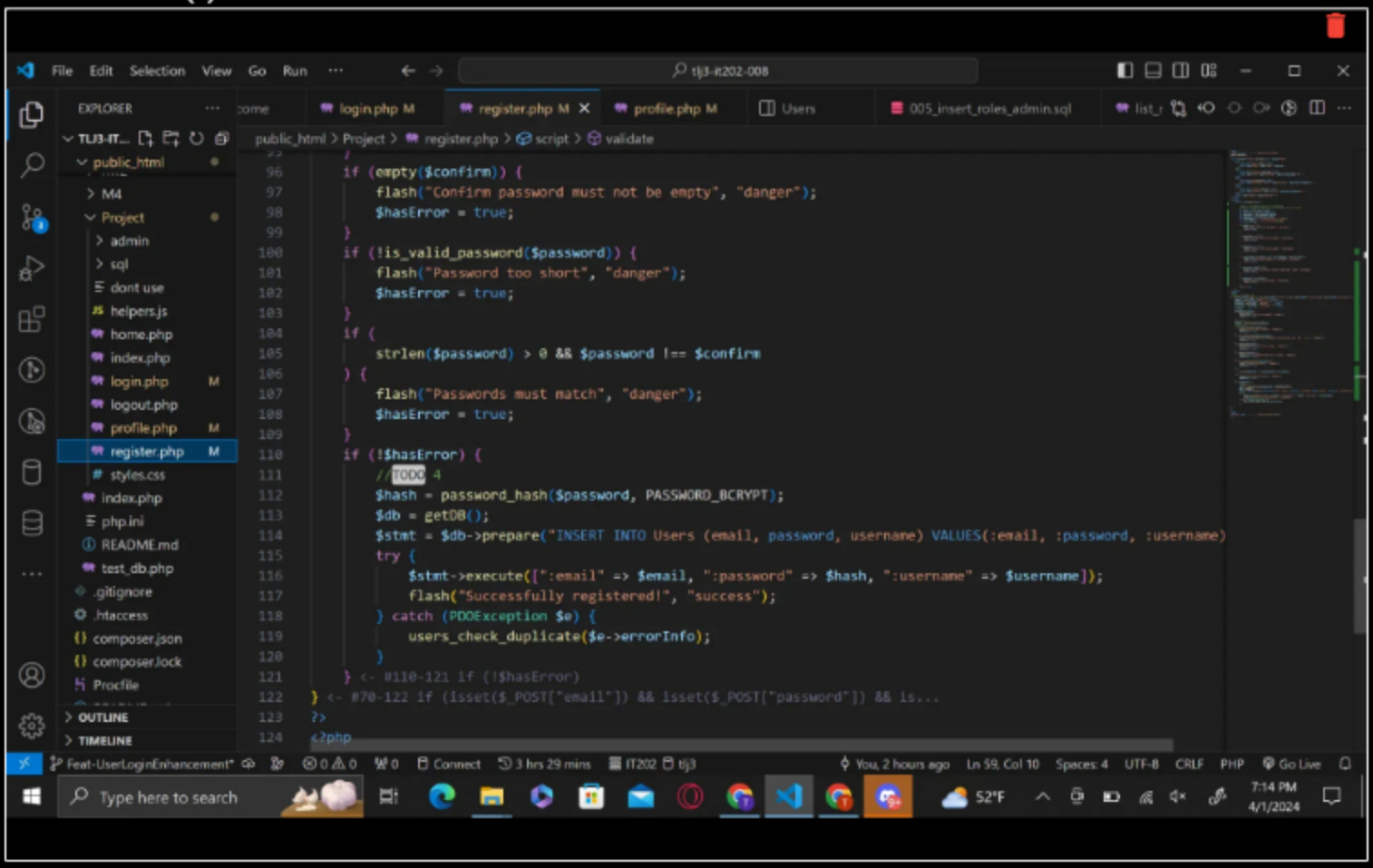
Checklist Items (0)





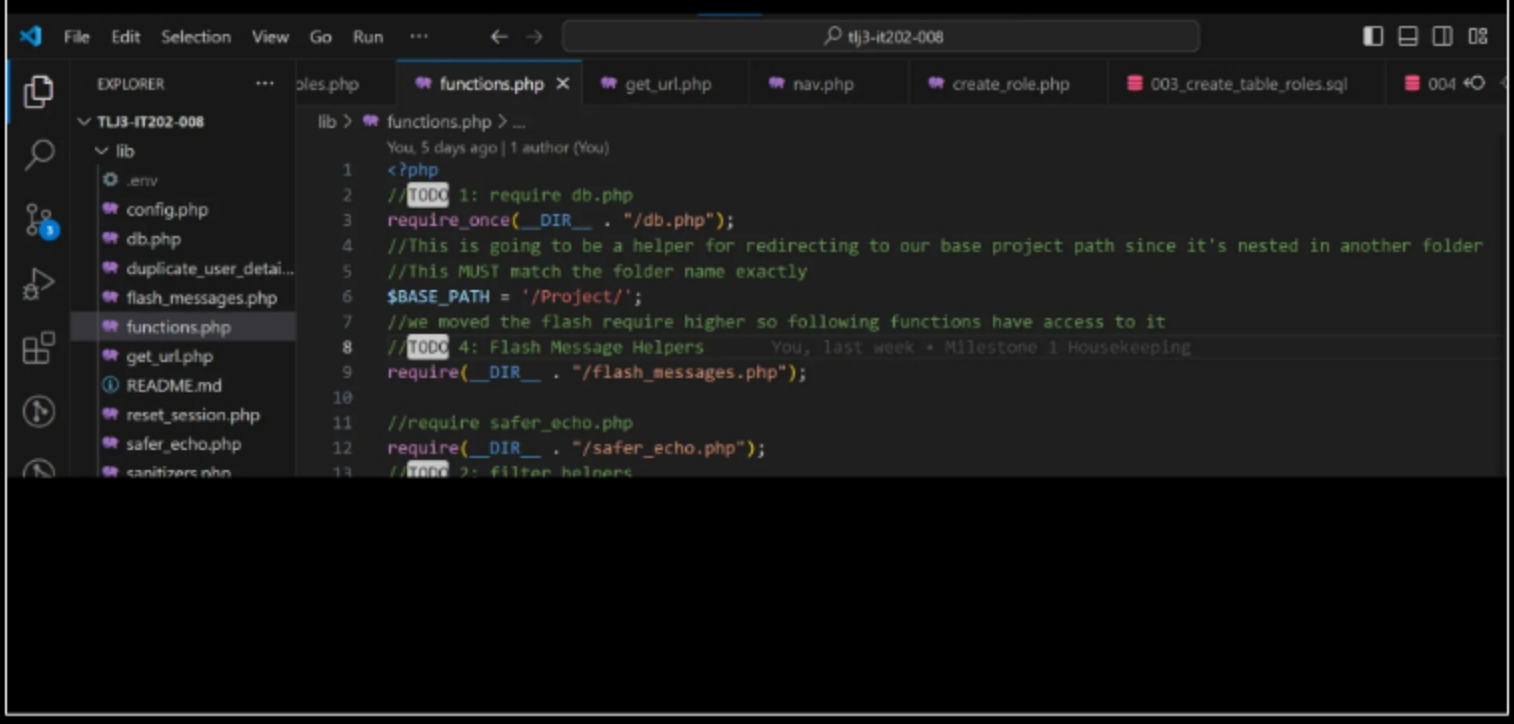
Part 1 of Server Side Validation

Checklist Items (0)



Part 2 of Server Side Validation

Checklist Items (0)



Function.php that was part of the user registration process.

Checklist Items (0)

COLLAPSE

Task #4 - Points: 1

Text: Screenshot of the Users table with a valid user entry

Checklist

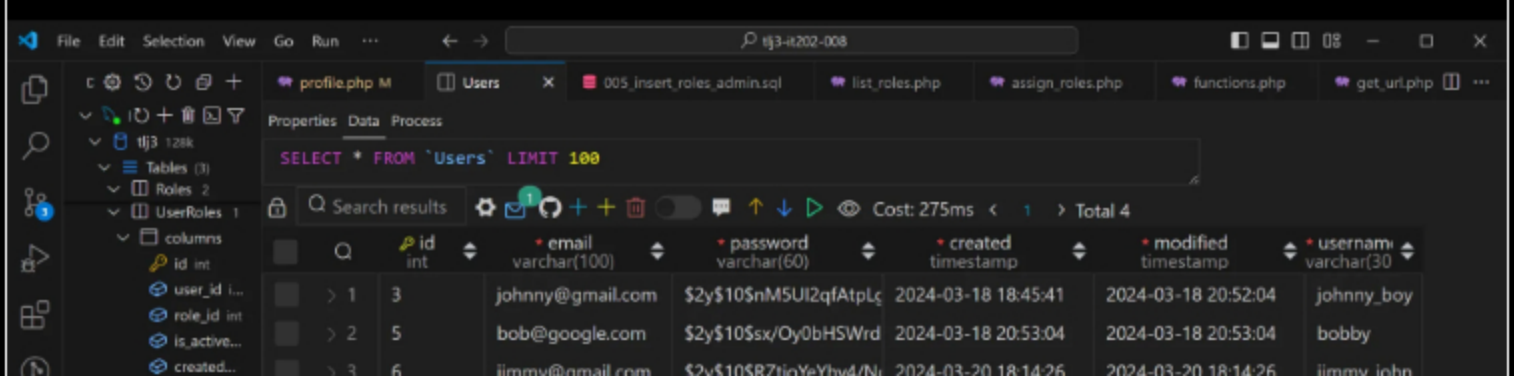
*The checkboxes are for your own tracking

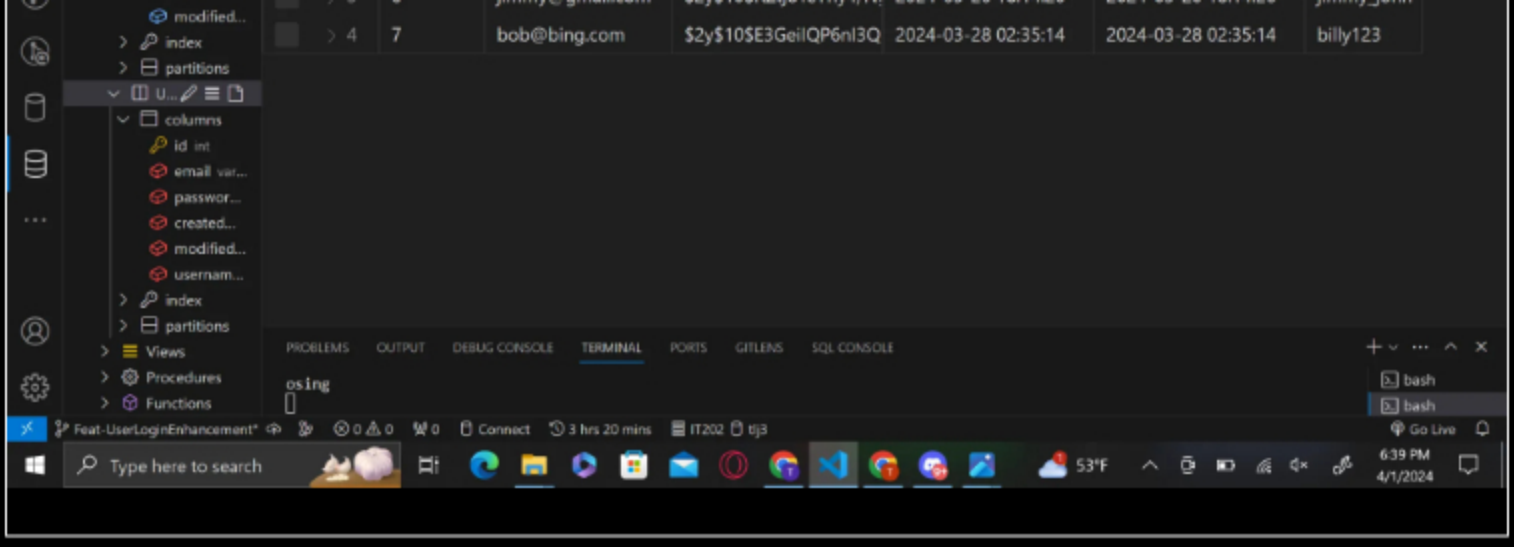
#	Points	Details
#1	1	Password should be hashed
#2	1	Should have email, password, username (unique), created, modified, and id fields
#3	1	Ensure left panel or database name is present (should contain your ucid)

Task Screenshots:

Gallery Style: Large View

Small Medium Large





This screenshot includes the User table, with the email, username, Id, password(hashed), created, and modified

Checklist Items (0)

COLLAPSE

Task #5 - Points: 1

Text: Explain the registration logic in a step-by-step manner from when the page loads to when the data is saved to the DB

Details:
Don't just show code, translate things to plain English

Response:

When the page loads, the user is left with a form that contains an email, username, password, and confirm password field. The user will fill those out while following the requirements and then submits it. The form will be sent to the server via an HTTP POST request. When its received, it goes though the server side validation to check for anything that doesn't meet the requirements (empty fields, valid email/user/password format, strong password, etc.). Once it passes the validation, the server interacts with the db to store the information from the user's form (including the username, email, password, id, etc.). Before the password is saved, it gets hashed using a hashing algorithm to ensure that the password isn't stored in plain text for enhanced security. Once the data is saved to the db, the server responds to the user with a success message.

COLLAPSE

Task #6 - Points: 1

Text: Include pull request links related to this feature

Details:
Should end in /pull/#

URL #1
https://github.com/Tiara-Jenks/tlj3-it202-008/pull/13

User Login (2 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Screenshot of form on website page

Checklist

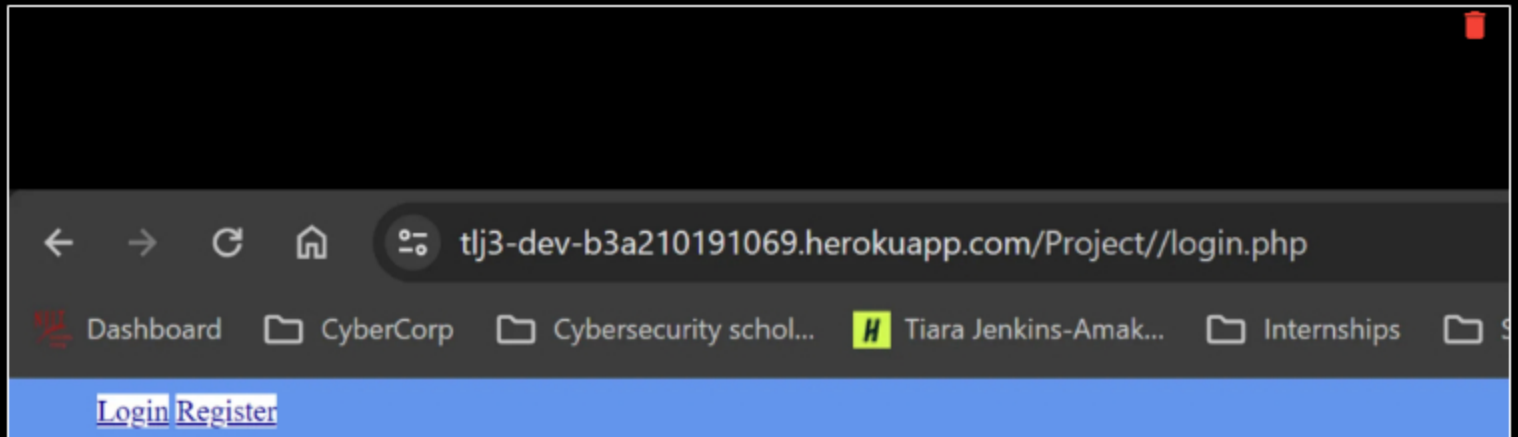
*The checkboxes are for your own tracking

#	Points	Details
#1	1	Heroku dev url should be present in the address bar
#2	1	Should have thoughtful CSS applied
#3	1	Include correct data where username is used to login
#4	1	Include correct data where email is used to login
#5	1	Show success login message
#6	1	Demonstrate JavaScript validation for each field [can be combined] (username format, email format, password format, and each field being required)
#7	1	Demonstrate user-friendly message of when an account doesn't exist
#8	1	Demonstrate user-friendly message of when password doesn't match what's in the DB
#9	1	Demonstrate successful login message and the destination page (i.e., home or some landing/dashboard page)
#10	1	Demonstrate session data being set (captured from server logs)

Task Screenshots:

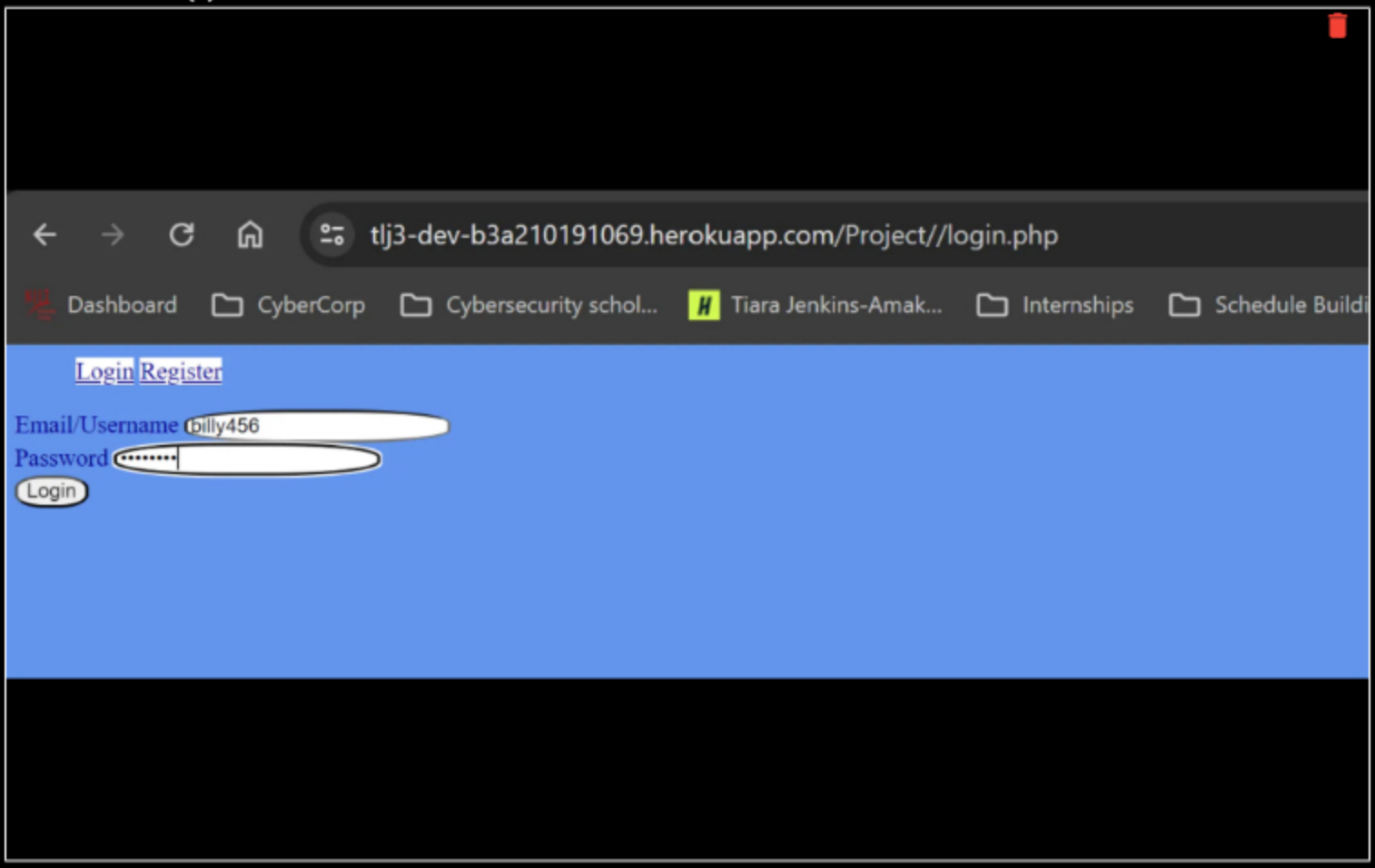
Gallery Style: Large View

Small Medium Large



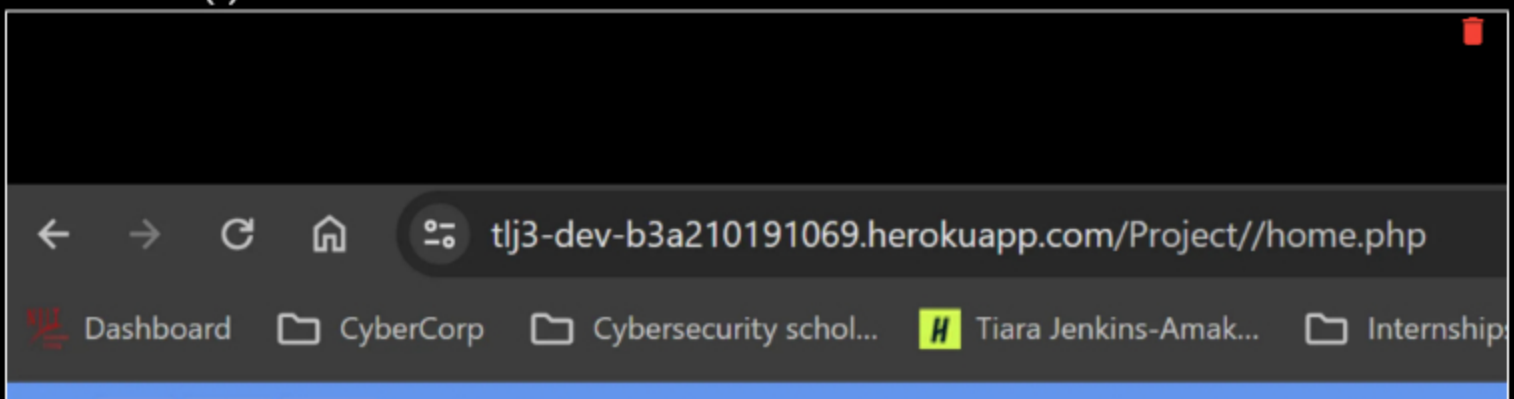
screenshot of the correct data where email is used to login

Checklist Items (0)



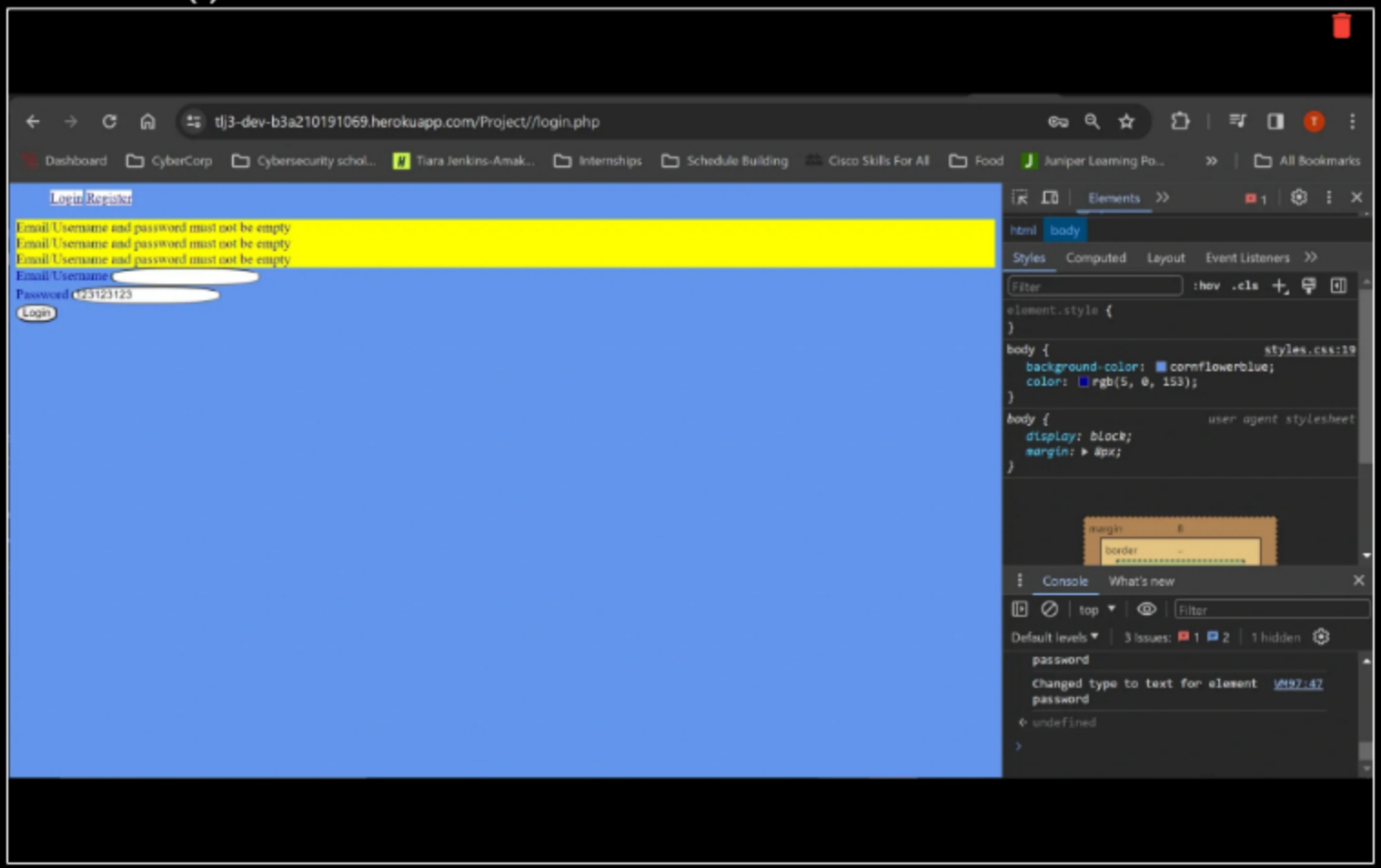
screenshot of the correct data where username is being used to login

Checklist Items (0)



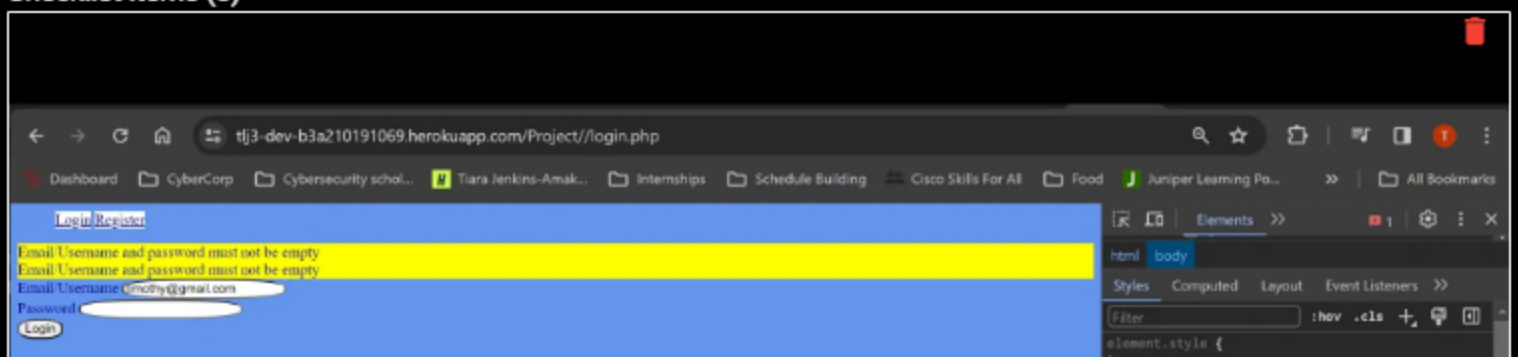
login success message

Checklist Items (0)



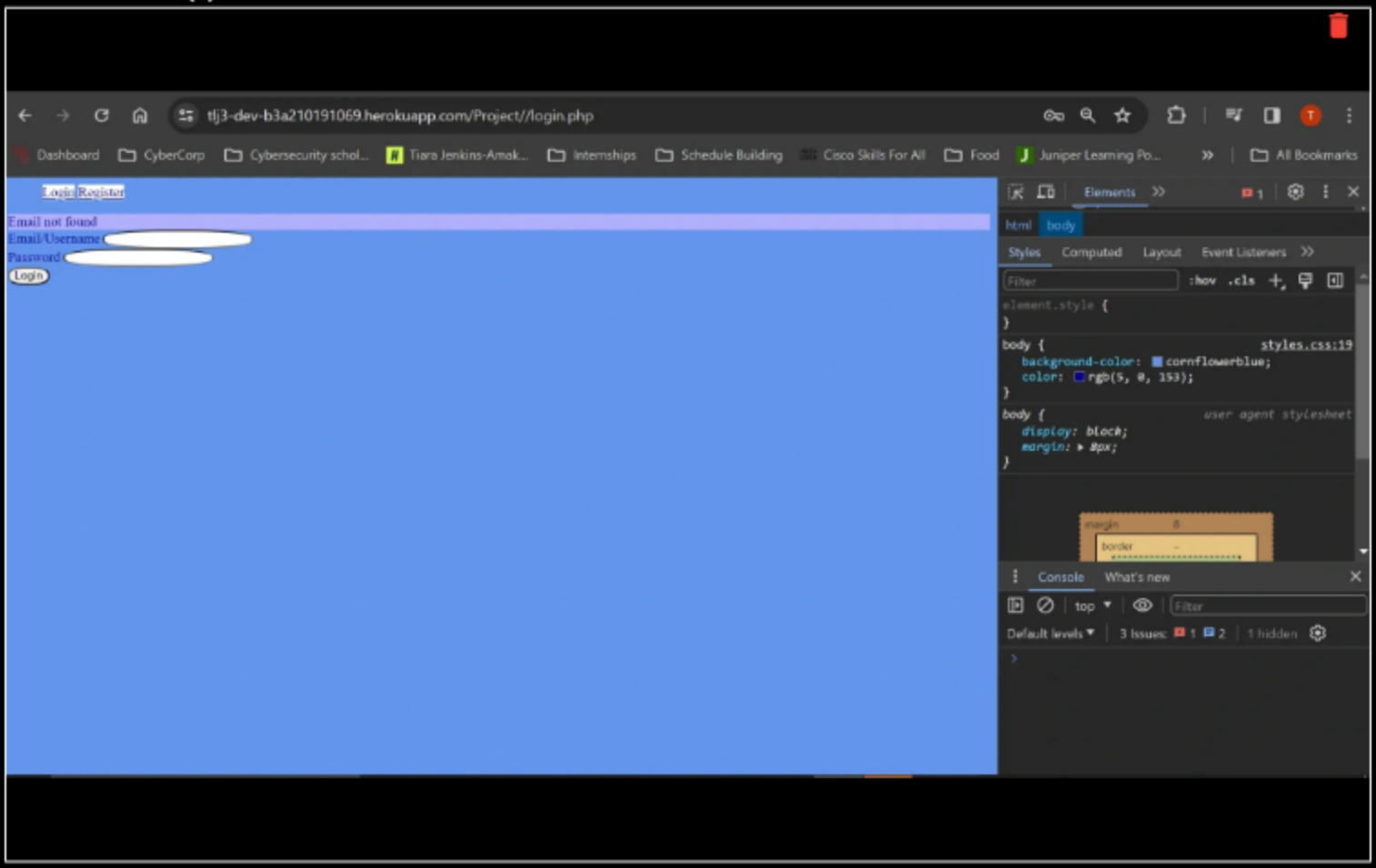
email and username are empty

Checklist Items (0)



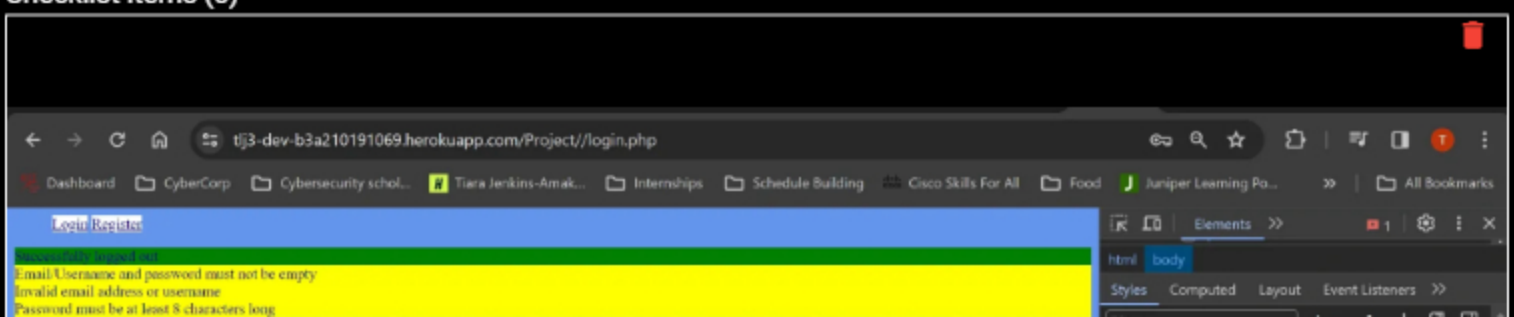
password is empty

Checklist Items (0)



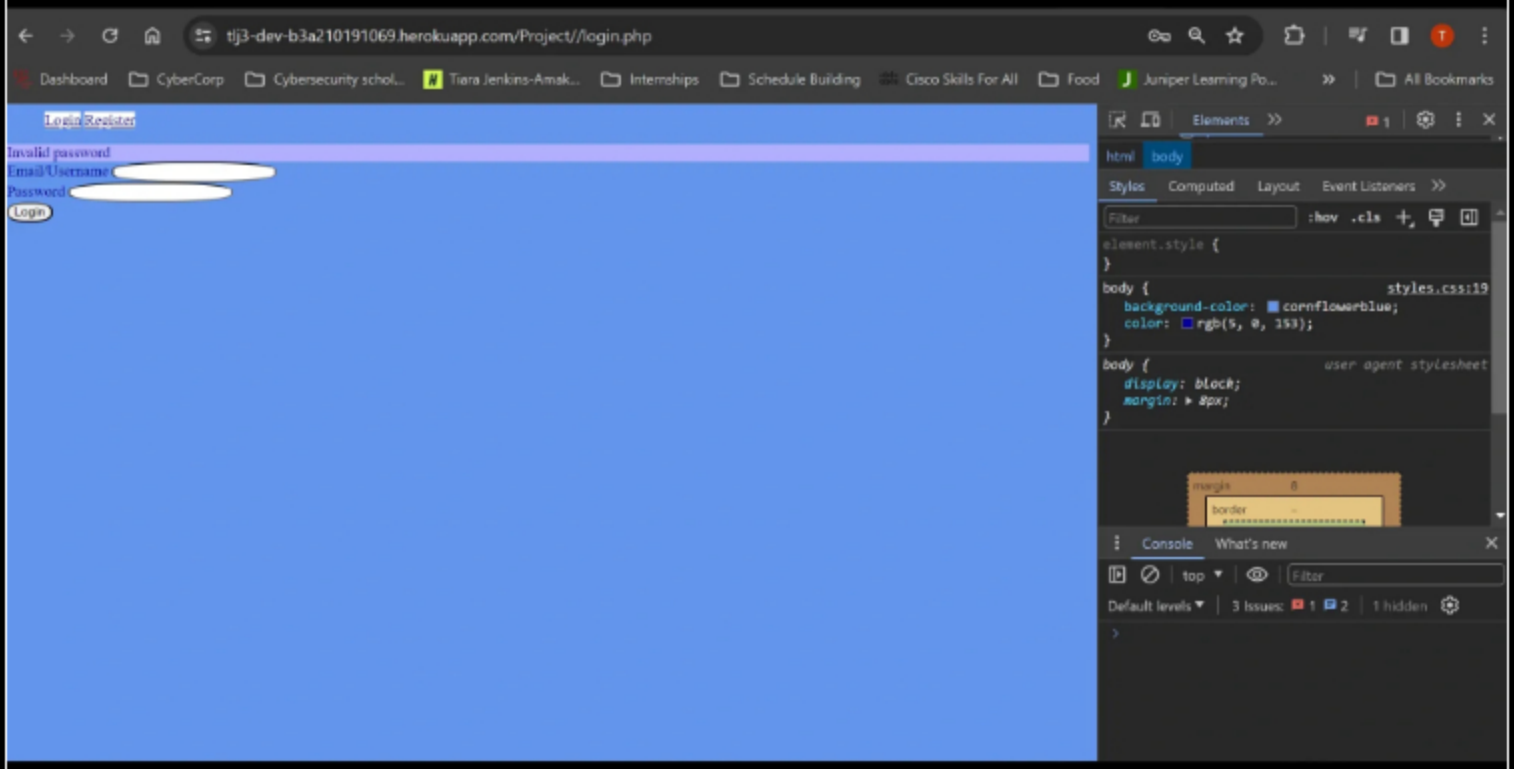
email not found

Checklist Items (0)



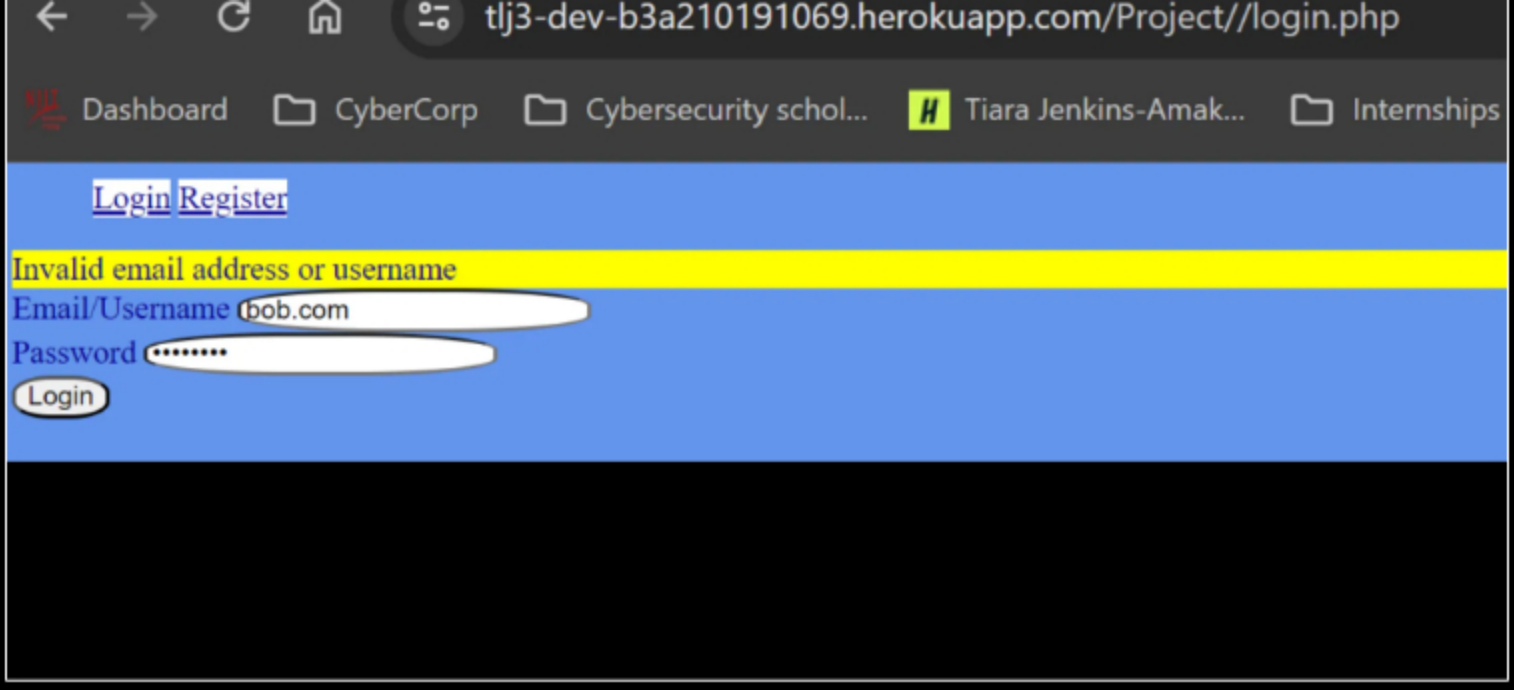
password doesn't meet requirements

Checklist Items (0)



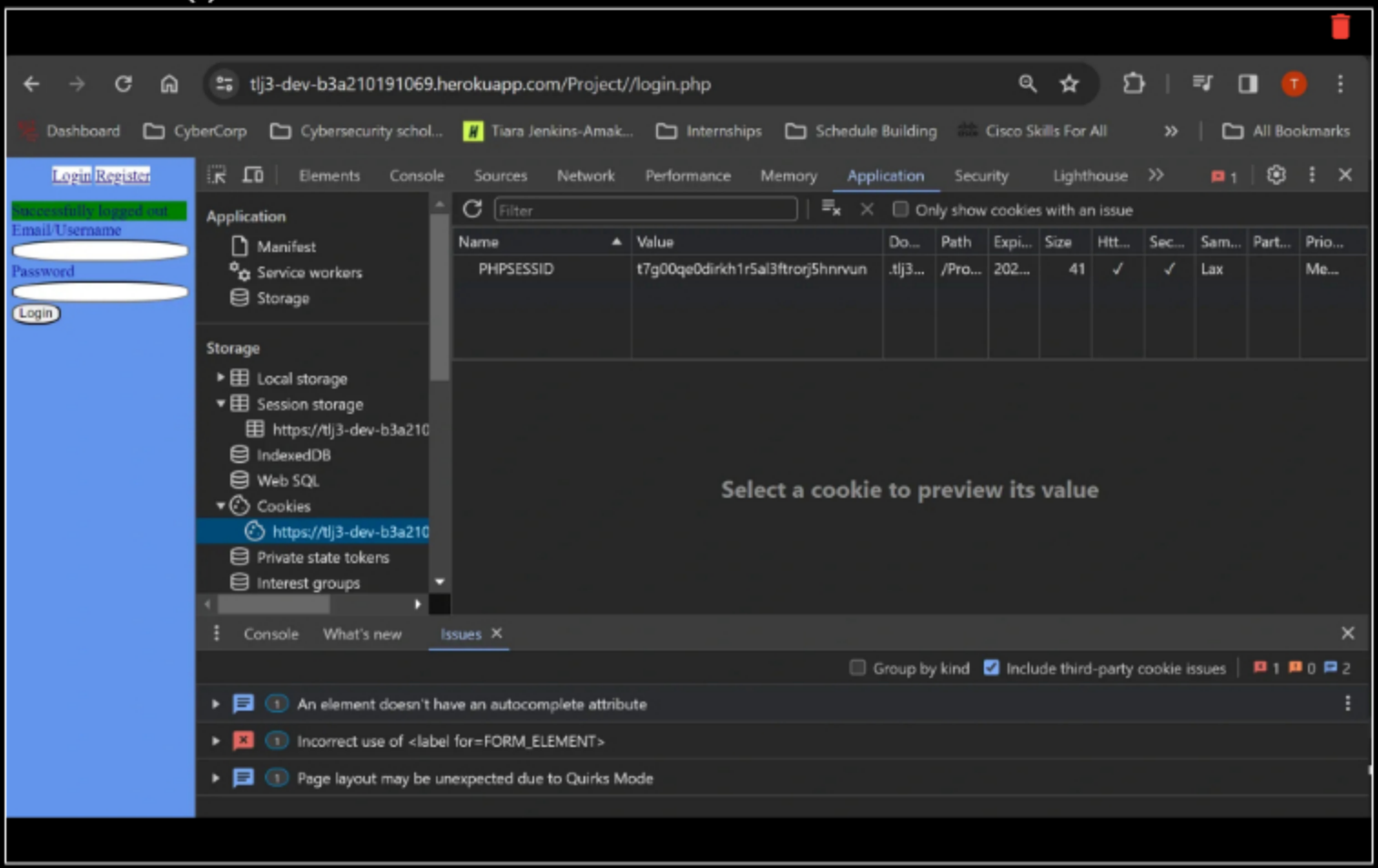
password didn't match the one in the db

Checklist Items (0)



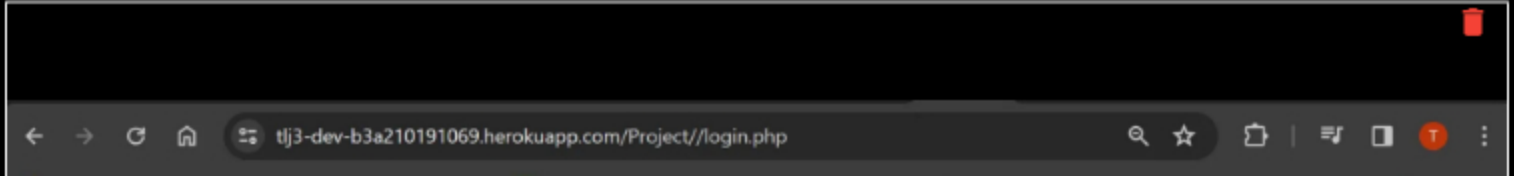
email and username are invalid as it doesn't meet the requirements

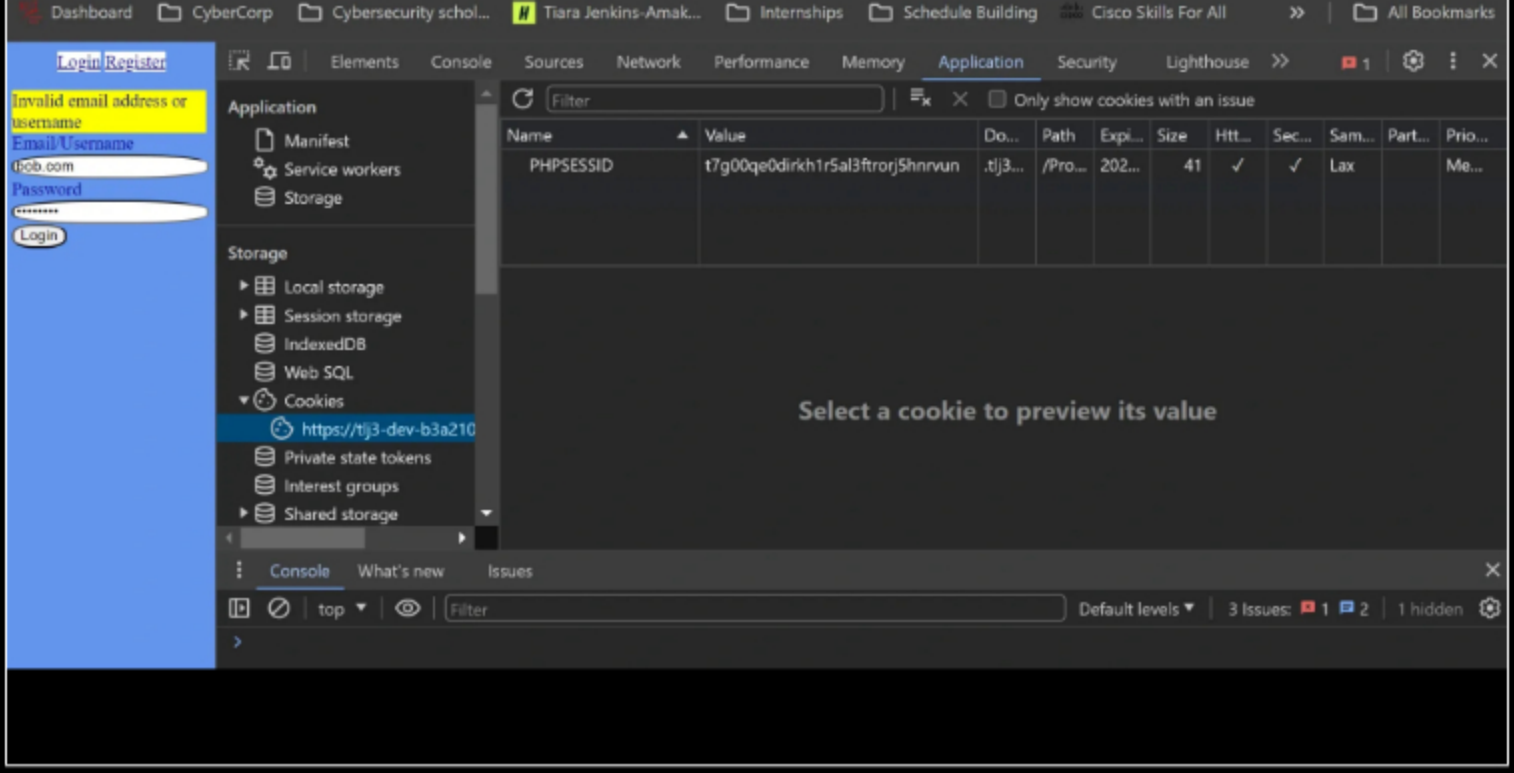
Checklist Items (0)



session data being captured (professor told me about the error_logs from home.php in application, but those are the only errors that I saw. If this isn't correct, please tell me in the notes on canvas or anywhere else)

Checklist Items (0)





this is a part 2 to the session data being set. I saw this in the Nav and login part 2 video on Learn and just posted this here to be on the safe side. Again, if this isn't correct, please tell me.

Checklist Items (0)

COLLAPSE

Task #2 - Points: 1

Text: Screenshot of the form code

Checklist

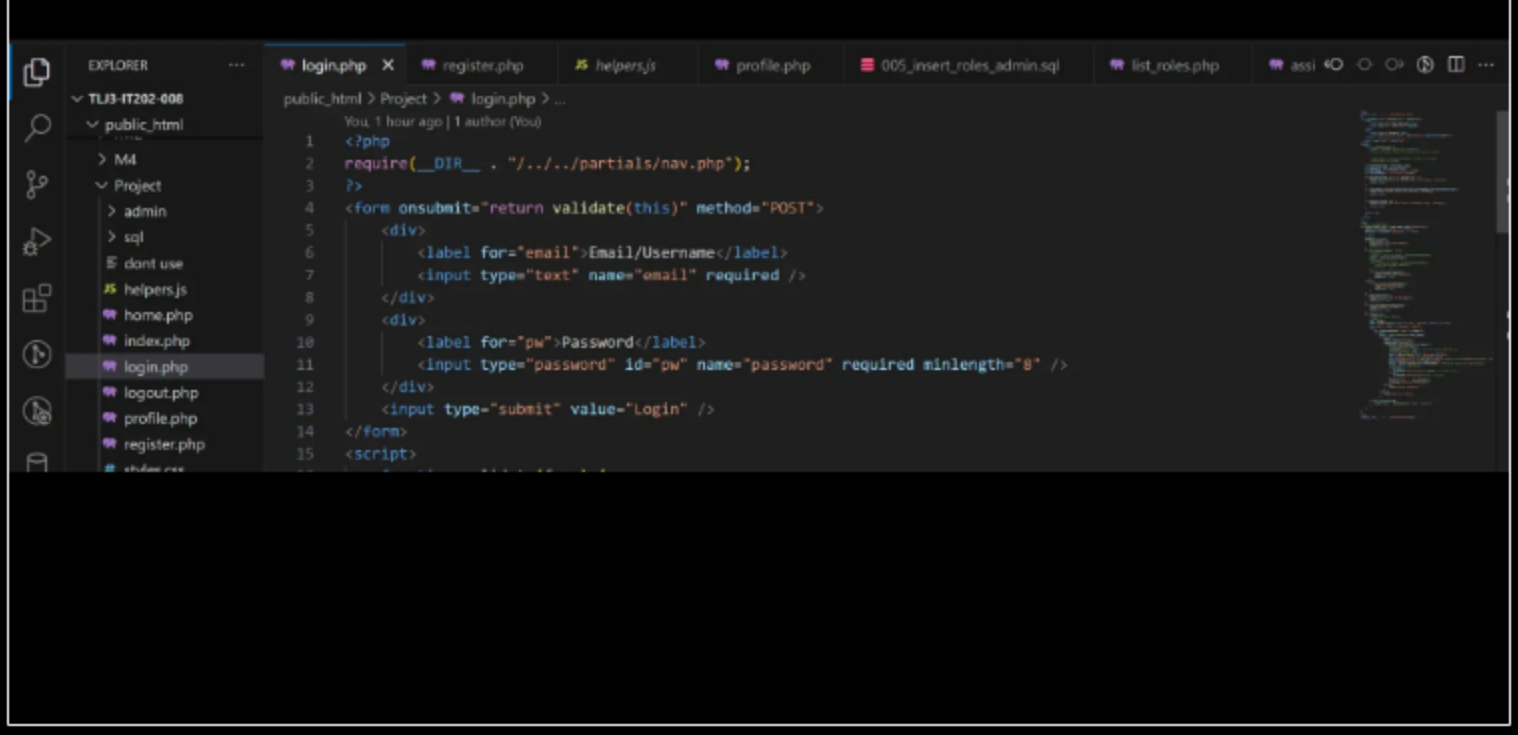
*The checkboxes are for your own tracking

#	Points	Details
#1	1	Should have proper input types for the fields (note in the caption if you're using two fields for Username/Email or a combined field)
#2	1	Show JavaScript validations (include any extra files related)
#3	1	Show PHP validations (include any lib content)
#4	1	Include ucid/date code comments in all screenshots (one per screenshot is sufficient)

Task Screenshots:

Gallery Style: Large View

Small Medium Large

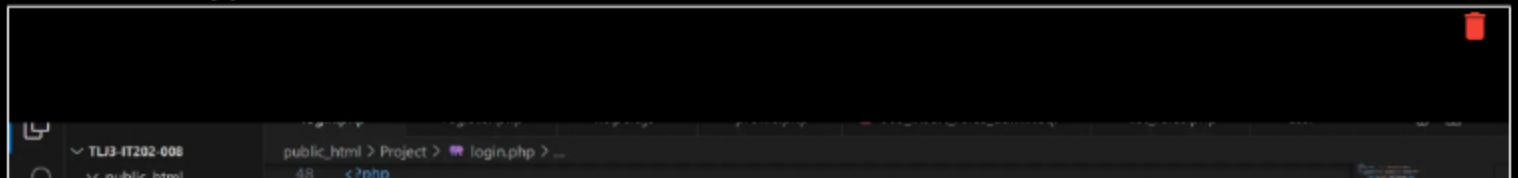


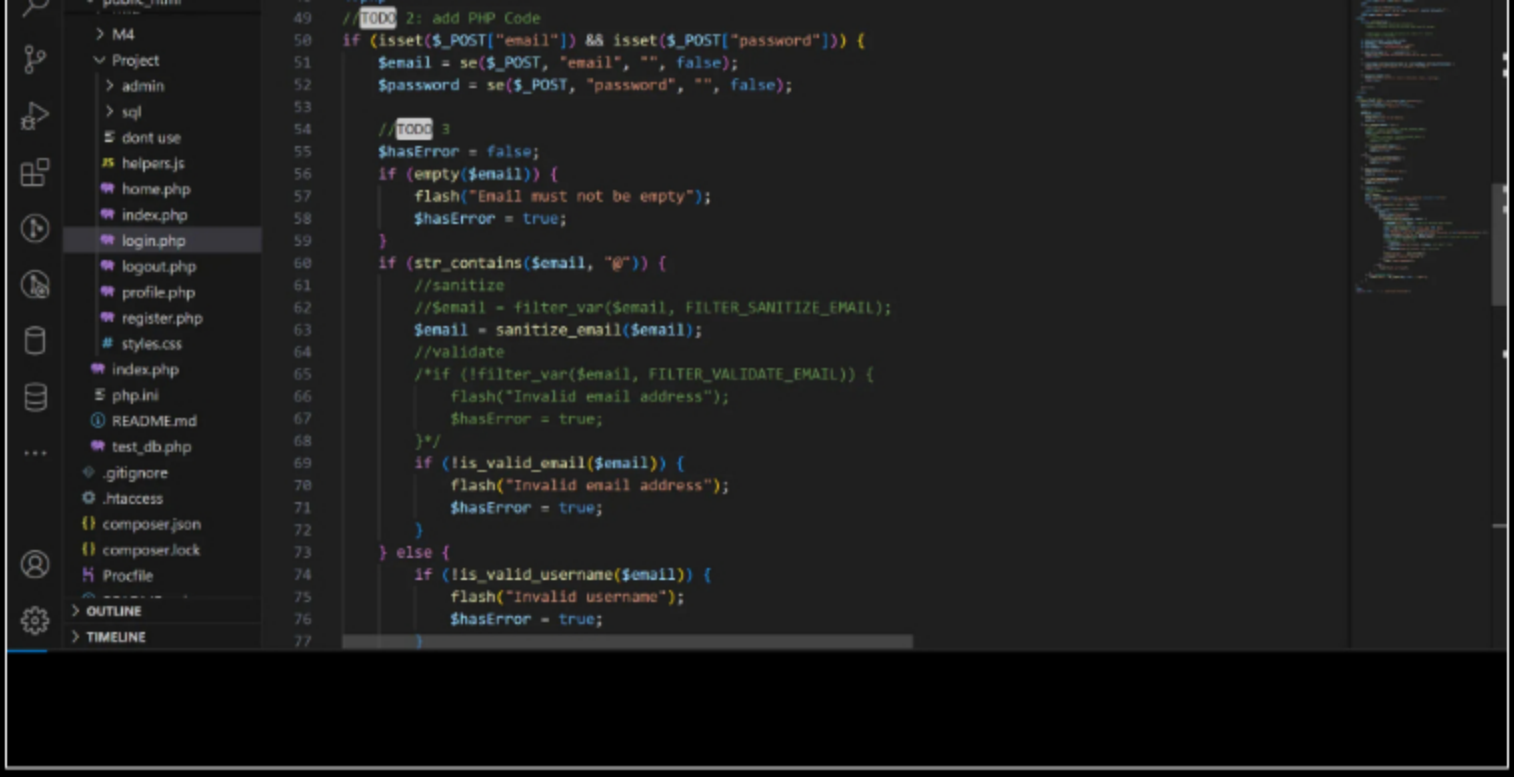
Screenshot of form code (used a combined field for email/username)

Checklist Items (0)

Screenshot of client side validation

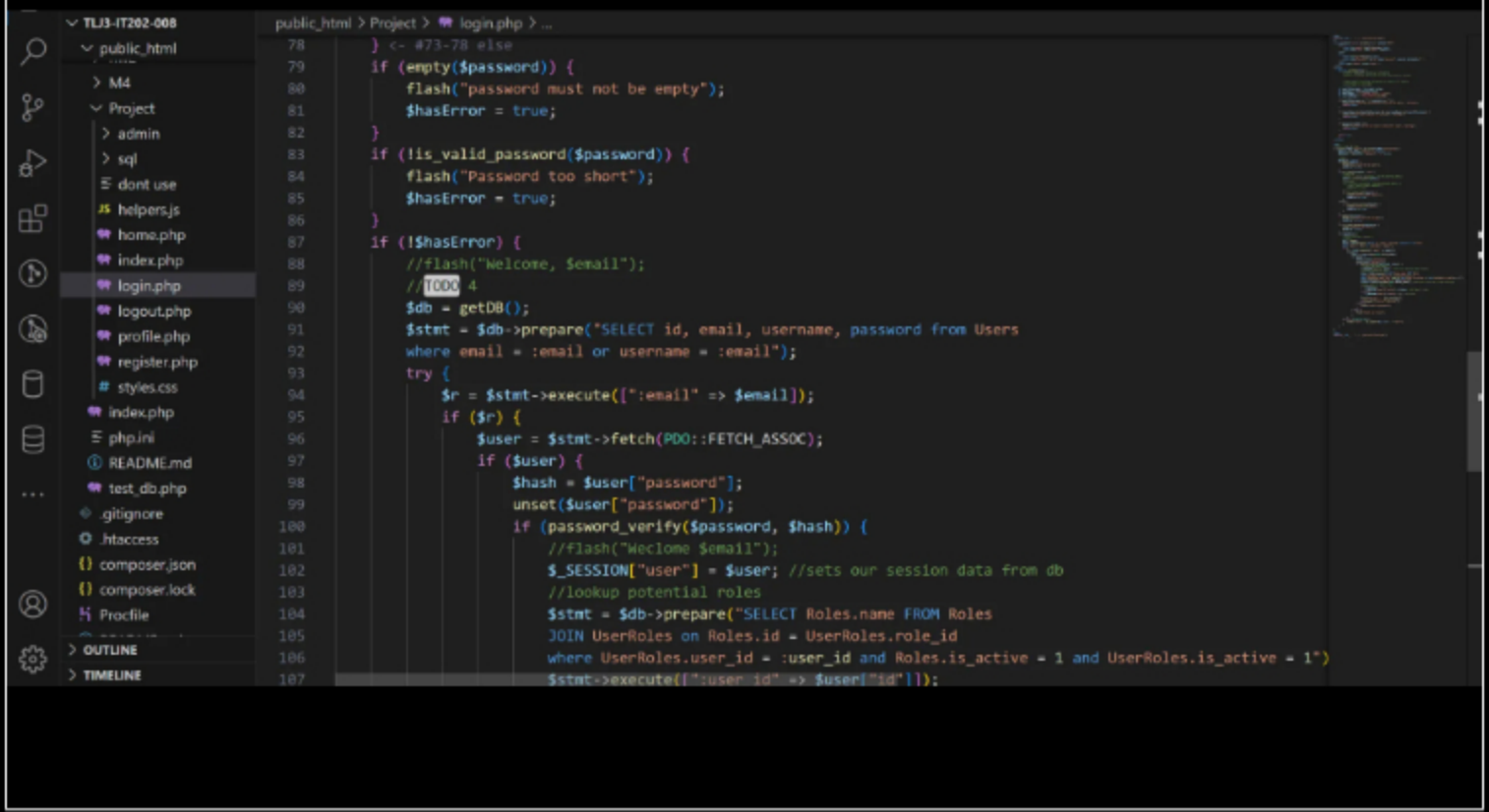
Checklist Items (0)





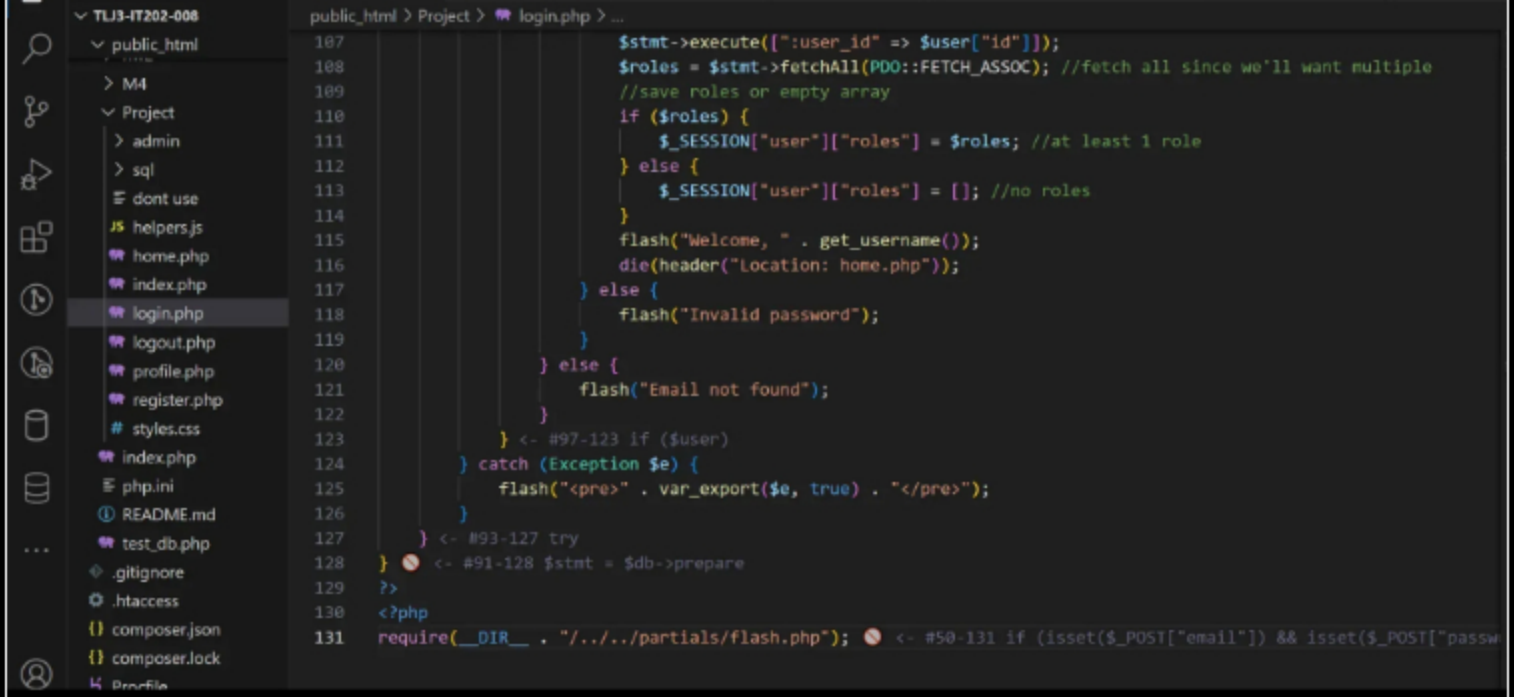
Screenshot of server side part1

Checklist Items (0)



Screenshot of server side part2

Checklist Items (0)



Screenshot of server side part3

Checklist Items (0)

COLLAPSE

Task #3 - Points: 1

Text: Explain the login logic step-by-step from when the page loads to when the data is fetched from the DB and stored in the session

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Don't just show code, translate things to plain English
#2	1	Explain how the session works and why/how it's used

Response:

When the page is loaded, the user is left with a form containing email/username and password fields. The user has to enter their credentials and submit the form. This data is sent to the server via HTTP POST request. Once received, it goes through the server side validation to ensure there's no empty fields, ensure the username/email exists within the db, and to ensure the password matches the one in the db. If the credentials pass the validation, the server interacts with the db to get the user's data based on the credentials provided (querying the users table to retrieve the user's records). Once the records are received, the server verifies the password submitted against the hashed password in the db, ensuring that they match. If this turns out successful, the server creates a new session for the user. The creation of a session involved generating a unique session ID and associating it with the data form the user (session ID is stored as a cookie).

COLLAPSE

Task #4 - Points: 1

Text: Include pull request links related to this feature

Details:
Should end in /pull/#

URL #1
https://github.com/TiaraJenks/tlj3-it202-008/pull/14

COLLAPSE

User Logout (1 pt.)

COLLAPSE

Task #1 - Points: 1

Text: Capture the following screenshots

Checklist

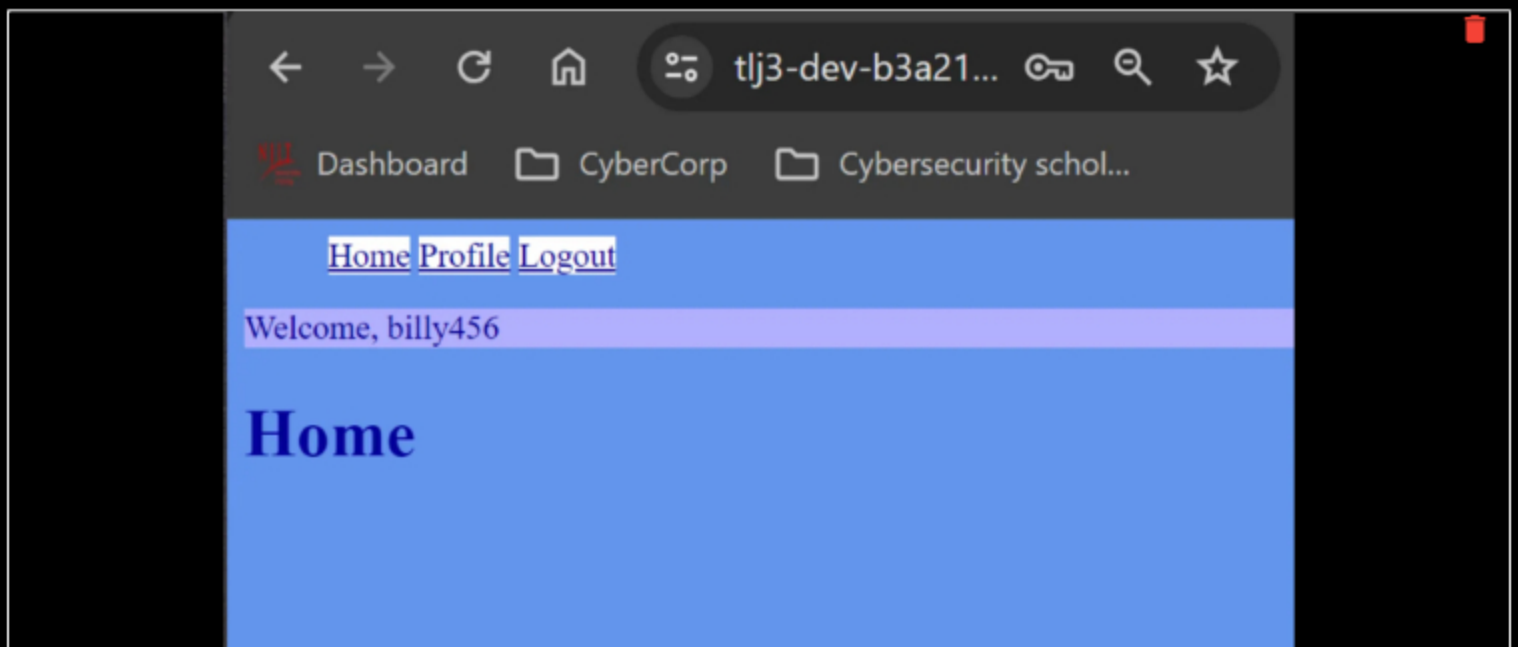
*The checkboxes are for your own tracking

#	Points	Details
#1	1	Screenshot of the navigation when logged in (site)
#2	1	Screenshot of the redirect to login with the user-friendly logged-out message (site)
#3	1	Screenshot of the logout-related code showing the session is destroyed (code). Ensure ucid/date comment is present.

Task Screenshots:

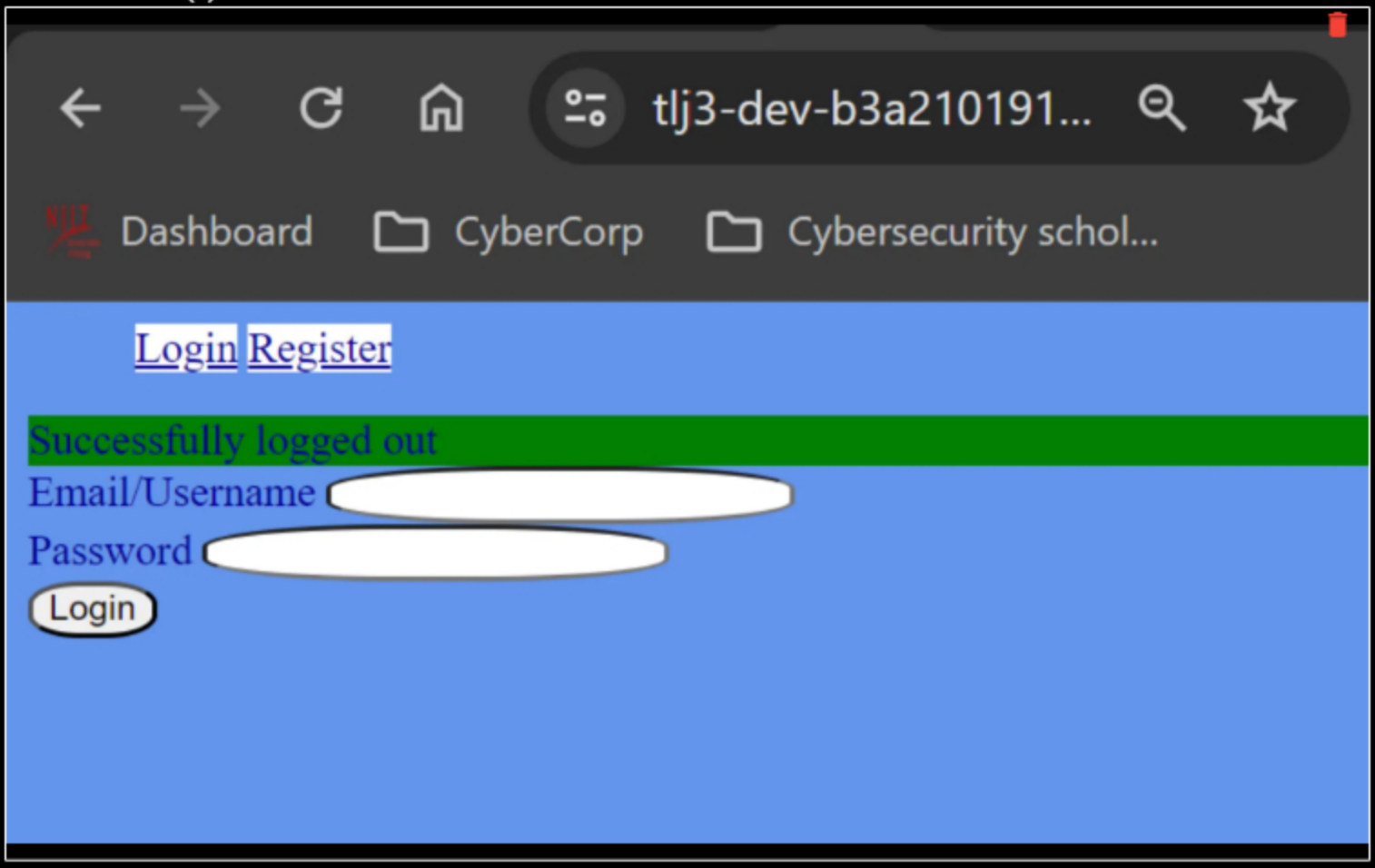
Gallery Style: Large View

Small Medium Large



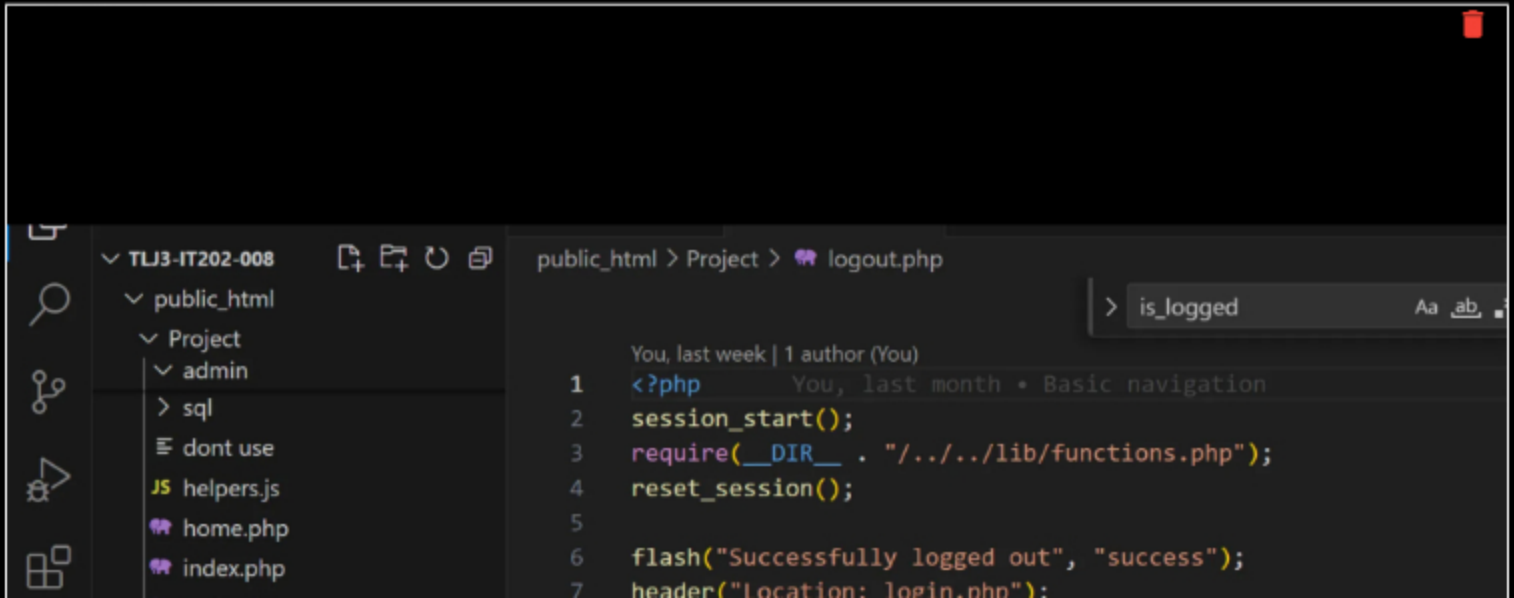
Navigation when logged in

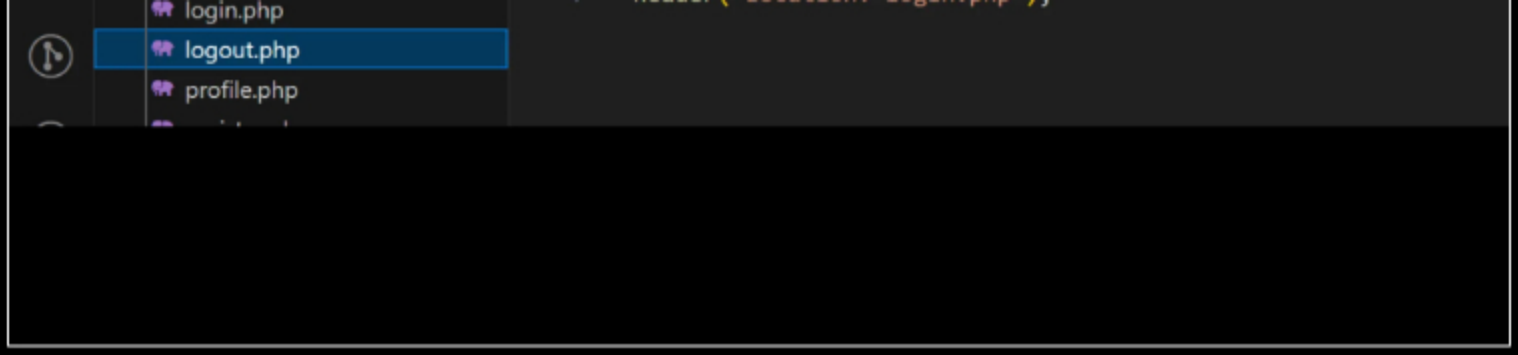
Checklist Items (0)



Logged out on site with user-friendly message

Checklist Items (0)





logout-related code showing the session is destroyed

Checklist Items (0)

COLLAPSE

Task #2 - Points: 1

Text: Include pull request links related to this feature

Details:
Should end in /pull/#

URL #1
https://github.com/TiaraJenks/tlj3-it202-008/pull/14

COLLAPSE

Basic Security Rules and Roles (2 pts.)

EXPAND

Task #1 - Points: 1

Text: Authentication Screenshots

EXPAND

Task #2 - Points: 1

Text: Authorization Screenshots

EXPAND

Task #3 - Points: 1

Text: Screenshots of UserRoles and Roles Tables

EXPAND

Task #4 - Points: 1

Text: Explain how Roles and UserRoles tables work in conjunction with the Users table

EXPAND

Task #5 - Points: 1

Text: Include pull request links related to this feature

COLLAPSE

User Profile (2 pts.)

EXPAND

Task #1 - Points: 1

Text: View Profile Website Page

EXPAND

Task #2 - Points: 1

Text: Explain the logic step-by-step of how the data is loaded and populated when the profile page is visited

EXPAND

Task #3 - Points: 1

Text: Edit Profile Website Page

EXPAND

Task #4 - Points: 1

Text: Explain the logic step-by-step of how the data is checked and saved for the following scenarios

EXPAND

Task #5 - Points: 1

Text: Include pull request links related to this feature

COLLAPSE

Misc (1 pt.)

EXPAND

Task #1 - Points: 1

Text: Screenshot of wakatime

EXPAND

Task #2 - Points: 1

Text: Screenshot of your project board from GitHub (tasks should be in the proper column)

EXPAND

Task #3 - Points: 1

Text: Provide a direct link to the project board on GitHub

Task #4 - Points: 1

EXPAND

Task #4 - Points: 1

Text: Provide a direct link to the login page from your prod instance

End of Assignment