

Course: IT202-010-S2025

Assignment: IT202 Milestone 1

Student: Justin C. (jjc88)

Status: Submitted | Worksheet Progress: 100%

Potential Grade: 10.00/10.00 (100.00%)

Received Grade: 0.00/10.00 (0.00%)

Grading Link: <https://learn.ethereallab.app/assignment/v3/IT202-010-S2025/it202-milestone-1/grading/jjc88>

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`

100%

Section #1: (2 pts.) Feature: User Will Be Able To Register A New Account

100%

Lecture 17 - Multithreading

Task #1 (0.6 / 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:

100%

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)

```
<form onsubmit="return validate(this)" method="POST">
  <div>
    <label for="email">Email</label>
    <input 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>
```

html code for reg

Login Register

Email

Username

Password ! Please fill out this field.

Confirm

email validation

Login Register

Email

Username

Password

Confirm ! Please fill out this field.

username val

Login Register

Email

Username

Password

Confirm ! Please fill out this field.

password fill val

Login Register

Email

Username

Password

Confirm ! Please lengthen this text to 8 characters or more (you are currently using 1 character).

password length val

Login Register

Email

Username

Password

Confirm

100%

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

```
function validateForm() {
    // This implements JavaScript validation
    // It returns false if there are errors and true for success
    let isValid = true;
    if (!isValidEmail(form.email.value)) {
        isValid = false;
        ElWarn("Please enter a valid email address.", "danger");
    }
    if (!isValidUsername(form.username.value)) {
        isValid = false;
        ElWarn("Username must be 3-30 characters and use only letters, numbers, _ or -.", "danger");
    }
    if (!isValidPassword(form.password.value)) {
        isValid = false;
        ElWarn("Password must be at least 8 characters long.", "danger");
    }
    if (form.password.value !== form.confirm.value) {
        isValid = false;
        ElWarn("Passwords do not match.", "danger");
    }
    if (!isValid) {
        tool_error("You'll see some red > uncommitted changes");
    }
}

// End of Listing
> --> JS code for reg validations
```

JS code for reg validations

Login Register

Please enter a valid email address.

Username must be 3-30 characters and use only letters, numbers, _ or -

Password must be at least 8 characters long

Passwords do not match

Email

Username

Password

Confirm

JS Validations all



Saved: 4/15/2025 4:25:17 PM

Text Prompt

Weight: 50%**Details:**

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

Your Response:

Using the example code given for password, I built off of that using validate. Created more functions in helper.js for other fields. Copied the requirements for each field from php to JS. Goes through each field with an IF statement.



Saved: 4/15/2025 4:25:17 PM

100%

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

in the instructions

- Include the outputs related to username/email already in use

```
if (empty($email)) {  
    $error[] = "Email must not be empty";  
}  
  
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {  
    $error[] = "Invalid email address";  
}  
  
if (empty($username)) {  
    $error[] = "Username must only contain 3-30  
    characters a-z, 0-9, __, or -";  
}  
  
if (empty($password)) {  
    $error[] = "password must not be empty";  
}  
  
if (empty($confirm)) {  
    $error[] = "Confirm password must not be empty";  
}  
  
if (empty($password) || empty($confirm)) {  
    $error[] = "Password too short";  
}
```

php code for reg validations



Login Register

Email must not be empty
Invalid email address
Username must only contain 3-30
characters a-z, 0-9, __, or -
password must not be empty
Confirm password must not be empty
Password too short

Email
Username
Password
Confirm

PHP Val pass and con empty



Login Register

Email must not be empty
Invalid email address
Username must only contain 3-30
characters a-z, 0-9, __, or -
Confirm password must not be empty
Password too short
Passwords must match

Email
Username
Password
Confirm

PHP Val password no match



Login Register

The chosen email is not available.
Email
Username
Password
Confirm

email taken



The chosen username is not available.

Email

Username

Password

Confirm

username taken



Saved: 4/15/2025 4:40:00 PM

100%

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/juicejoose/jjc88-it202-010-Milestone1/public_html/project/register.php



URL

https://github.com/juicejoose/jjc88-it202-010-Milestone1/public_html/project/register.php



URL #2

<https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com/project/register.php>



URL

<https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com/project/register.php>



URL #3

<https://github.com/juicejoose/jjc88-it202-010/pull/4>



URL

<https://github.com/juicejoose/jjc88-it202-010/pull/4>



Saved: 4/15/2025 4:50:25 PM

100%

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 (preferably a recent one during evidence capturing)

user data base



Saved: 4/15/2025 4:42:15 PM

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

100%

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)

```
form onsubmit="return validate(this)" method="POST">
  <div>
    <label for="email">Email/Username</label>
    <input type="text" name="email" required />
  </div>
  <div>
    <label for="pw">Password</label>
    <input type="password" id="pw" name="password" required minlength="8" />
  </div>
  <input type="submit" value="Login" />
</form>
```

html code for login

The screenshot shows a light blue-themed login/register form. At the top, there are two input fields: 'Email/Username' and 'Password'. Below these are two buttons: 'Login' and 'Register'. A validation message 'Please fill out this field.' is displayed in a white box with a red border and an exclamation mark icon, pointing to the 'Email/Username' field. The background has horizontal stripes.

html email/user fill val

Login Register

Email/Username

Password

 Please fill out this field.

html password fill val

Login Register

Email/Username

Password

 Please lengthen this text to 8 characters or more (you are currently using 1 character).

html password length val

 Saved: 4/15/2025 4:53:30 PM

Text Prompt

Weight: 50%

Details:

100%

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

```


script:
function validate(form) {
    //TODO : implement JavaScript validation
    //ensure it returns false for an error and true for success
    //TODO update clientside validation to check if it should
    //valid email or username
    //jjc88 4/14/2025 JS Validation for email and password
    let isValid = true;
    if (!isValidEmail(form.email.value)) {
        isValid = false;
        flash("Please enter a valid email address or username.", "danger");
    }
    if (!isValidPassword(form.password.value)) {
        isValid = false;
        flash("Password must be at least 8 characters long", "danger");
    }
    return true;
}
-- #16-35 function validate(form)


```

JS code for login validations

The screenshot shows a form titled "Login Register". It has two red validation messages: "Please enter a valid email address or username." and "Password must be at least 8 characters long". The "Email/Username" field contains "awdawd" and the "Password" field is empty. A "Login" button is visible below the fields.

JS Validation all

Saved: 4/15/2025 4:55:13 PM

Text Prompt

Weight: 50%

Details:

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

Your Response:

Using the example code given for password, I built off of that using validate. Created more functions in helper.js for other fields. Copied the requirements for each field from php to JS. Goes through each field with an IF statement. Code is very similar to reg page.

100%

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

```

$ cd /opt/ctf/level10
$ ./check.sh
[+] Email must not be empty
[+] Invalid username
[+] Password must not be empty
[+] Password too short
[+] Email/Username
[+] Password
[+] Login

```

PHP code for login validations

Login Register

Email must not be empty
Invalid username
password must not be empty
Password too short

Email/Username

Password

PHP Validations

Login Register

Email not found

Email/Username

Password

PHP email not found

Login Register

Invalid password

Email/Username

Password

PHP invalid password



Saved: 4/15/2025 5:03:26 PM

≡ Text Prompt

Weight: 50%

Details:

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

100%

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	 URL https://github.com/juicejoose/jjc88-it202-010/Milestone1/public_html/project/login.php	
URL #2	 URL https://jjc88-it202-010-prod-b44e1de4e1de4d69c.herokuapp.com/project/login.php	
URL #3	 URL https://github.com/juicejoose/jjc88-it202-011076	

Saved: 4/15/2025 5:05:50 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

```
2020-04-15T21:12:01.745544+00:00 app[web.1]: 10.1.23.125 - - [15/Apr/2020:21:12:01 +0000] "POST /project/test5.php HTTP/1.1" 302 3616 {"host": "https://11e605-1e200-010 dev- c9108f4d-4848-43c5-bc0a-80a5a5057-36 (KHTML, like Gecko) Chrome/105.0.5195.102 Safari/537.36 (KHTML, like Gecko) Chrome/105.0.5195.102 Safari/537.36 (KHTML, like Gecko) Chrome/105.0.5195.102 Safari/537.36"}
```

Heroku error output log

 Saved: 4/15/2025 5:14:02 PM

100%

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

100%

Task #1 (1 pt.) - Evidence

Combo Task:

Weight: 100%

Objective: Evidence

Image Prompt

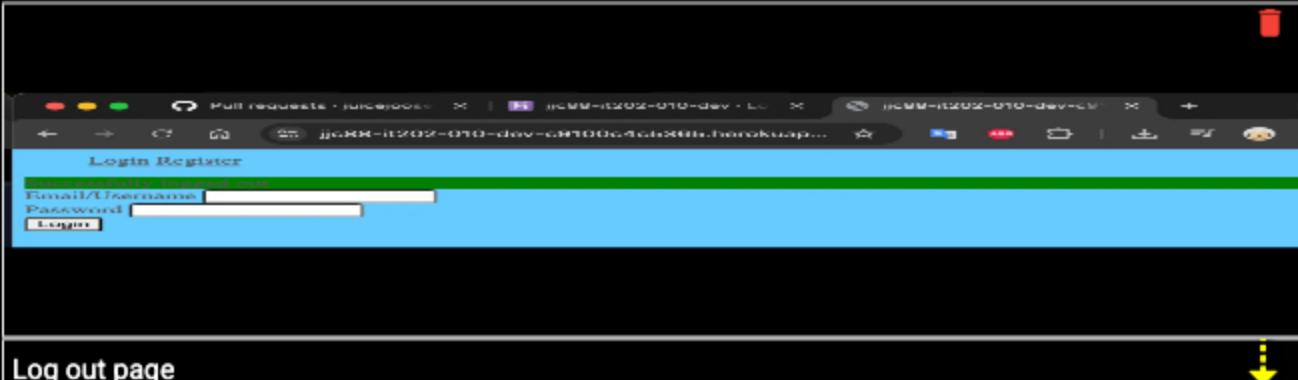
Weight: 40%

Details:

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

```
1 <?php
2 session_start();
3 require(__DIR__ . "/../../lib/functions.php");
4 reset_session();
5
6 flash("Successfully logged out", "success");
7 header("Location: login.php");| Justin Chung, Te
```

logout code





Saved: 4/15/2025 5:16:56 PM

Url Prompt

Weight: 20%

Details:

- Include the pull request url for this feature

URL #1

<https://github.com/juicejoose/jjc88-it202p0107>



URL

<https://github.com/juicejoose/jjc88>



Saved: 4/15/2025 5:16:56 PM

Text Prompt

Weight: 40%

Details:

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

Your Response:

when the logout.php is called, a reset_sessions is used to clear the session and data associated. It'll then flash a message after the session resets stating Successfully logged out, and then redirects them back to login.



Saved: 4/15/2025 5:16:56 PM

100%

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

100%

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

The screenshot shows the MySQL Workbench interface with the 'Roles' table selected. The table has columns: id (int), role_name (varchar), is_active (tinyint), created_timestamp (timestamp), and modified_timestamp (timestamp). There is one row with data: id=1, role_name='Admin', is_active=1, created_timestamp='2023-01-01 12:00:00', modified_timestamp='2023-01-01 12:00:00'. A red trash can icon is in the top right corner.

Roles data

The screenshot shows the MySQL Workbench interface with the 'UserRoles' table selected. The table has columns: user_id (int), role_id (int), and created_timestamp (timestamp). There is one row with data: user_id=1, role_id=1, created_timestamp='2023-01-01 12:00:00'. A red trash can icon is in the top right corner.

100%

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

```
if (!has_role("Admin")) {  
    flash("You don't have permission to view this page", "warning");  
    die(header("Location: $BASE_PATH" . "/home.php"));  
}
```

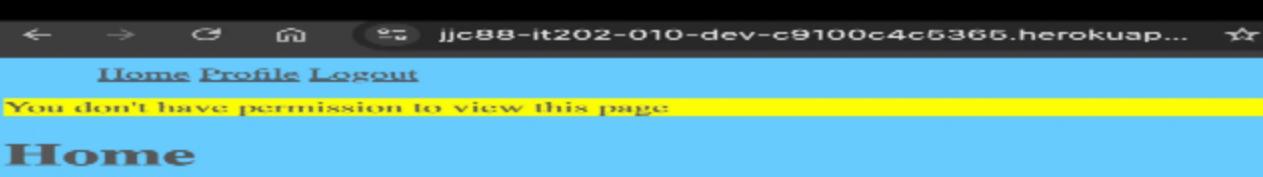
role check

```
if (is_logged_in(true)) {  
    //comment this out if you don't want to see the session variables  
    error_log("Session data: " . var_export($_SESSION, true));  
}  
?>
```

login check



Need to be logged in page



proper permission page



Url Prompt

Weight: 20%

Details:

- Include the pull request url for this feature

URL #1

<https://github.com/juicejoose/jjc88-it202501015>



URL

<https://github.com/juicejoose/jjc88-it202501015>



Saved: 4/15/2025 5:35:27 PM

Text Prompt

Weight: 40%

Details:

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

100%

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

100%

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:

100%

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)

```
<form method="POST" onsubmit="return validate(this);">
    <div class="mb-3">
        <label for="email">Email</label>
        <input type="email" name="email" id="email" value="7.php@se(Semantic).7.ee" />
    </div>
    <div class="mb-3">
        <label for="username">Username</label>
        <input type="text" name="username" id="username" value="7.php@se(Semantic).7.ee" />
    </div>
    <div>DO NOT PRELOAD PASSWORD</div>
    <div>Password Reset</div>
    <div class="mb-3">
        <label for="cp">Current Password</label>
        <input type="password" name="currentPassword" id="cp" />
    </div>
    <div class="mb-3">
        <label for="np">New Password</label>
        <input type="password" name="newPassword" id="np" />
    </div>
    <div class="mb-3">
        <label for="cpn">Confirm Password</label>
        <input type="password" name="confirmPassword" id="cpn" />
    </div>
    <input type="submit" value="Update Profile" name="submit" />
</form>
```

html code for profile



html email val



Saved: 4/15/2025 8:07:02 PM

Text Prompt

Weight: 50%

Details:

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

Your Response:

POST is used to update the profile data with new values. Each field is associated with its type to avoid mismatch. The values of some fields will be already filled from previous data.



Saved: 4/15/2025 8:07:02 PM

100%

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



```
function validateProfile() {
    var email = document.getElementById("Email").value;
    var username = document.getElementById("Username").value;
    var currentPassword = document.getElementById("Current Password").value;
    var newPassword = document.getElementById("New Password").value;
    var confirmPassword = document.getElementById("Confirm Password").value;

    if (!email) {
        alert("Email is required");
        return false;
    }

    if (!username) {
        alert("Username is required");
        return false;
    }

    if (!newPassword) {
        alert("New password is required");
        return false;
    }

    if (newPassword.length < 8) {
        alert("New password must be at least 8 characters");
        return false;
    }

    if (newPassword !== confirmPassword) {
        alert("New password and Confirm password must match");
        return false;
    }

    if (currentPassword !== newPassword) {
        alert("Current password does not match new password");
        return false;
    }

    return true;
}
```

JS code for profile validations

Enter a valid email
Username must be 3-30 characters:
letters, numbers, _ or -
New password must be at least 8
characters
Password and Confirm password must
match

Email

Username

Password Reset

Current Password

New Password

Confirm Password

JS profile val all



Saved: 4/15/2025 8:03:39 PM

Text Prompt

Weight: 50%

Details:

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

Your Response:

In profile bunch of variables are made to link to their values associated. Using if statements to validate data, for email making sure certain characters and will flash if flagged. Same for username. For password, it'll check it current is same and checking password requirements such as length.



Saved: 4/15/2025 8:03:39 PM

100%

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

Combo Task:

Weight: 33.33%

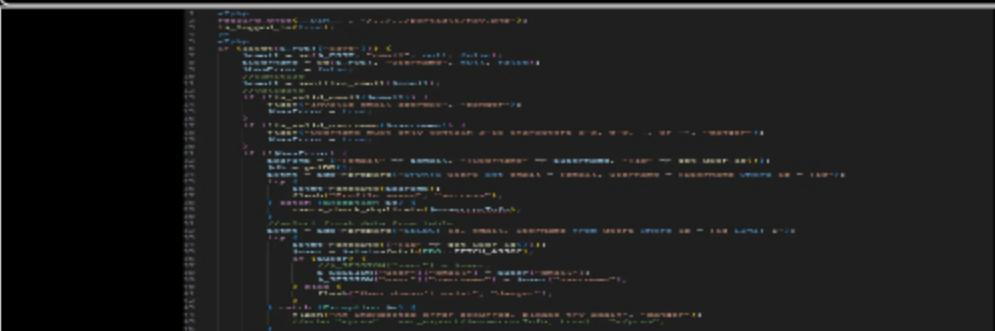
Objective: PHP Validation (steps before the DB call)

Image Prompt

Weight: 50%

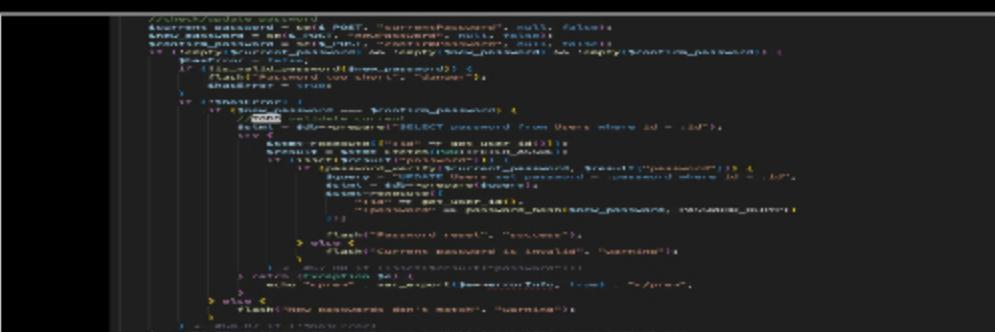
Details:

- 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



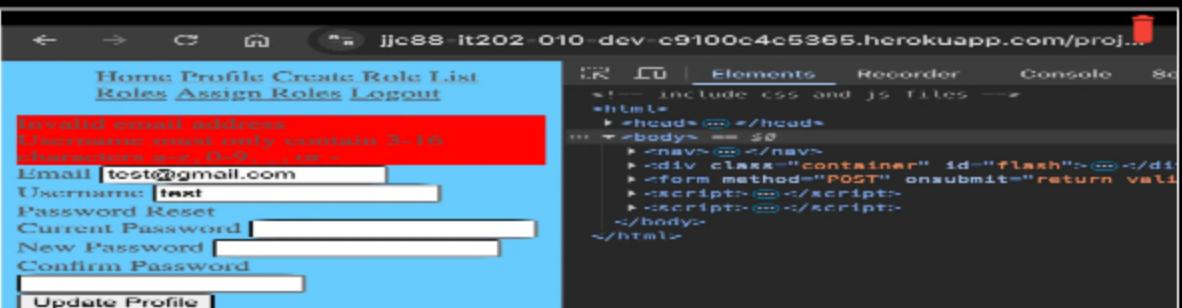
```
if($username == '') {
    $error[] = "Username must contain at least one character";
}
if($email == '') {
    $error[] = "Email must contain at least one character";
}
if($password == '') {
    $error[] = "Password must contain at least one character";
}
if($confirm_password == '') {
    $error[] = "Confirm Password must contain at least one character";
}
if($username != '' & !filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $error[] = "Email must be a valid email address";
}
if($password != $confirm_password) {
    $error[] = "Passwords do not match";
}
if($username != '' & $username != 'admin') {
    $query = "SELECT * FROM users WHERE username = '$username'";
    $result = mysqli_query($conn, $query);
    if(mysqli_num_rows($result) > 0) {
        $error[] = "Username is already taken";
    }
}
```

php code 1



```
if($current_password == '') {
    $error[] = "Current Password must contain at least one character";
}
if($new_password == '') {
    $error[] = "New Password must contain at least one character";
}
if($confirm_new_password == '') {
    $error[] = "Confirm New Password must contain at least one character";
}
if($current_password != $new_password) {
    $error[] = "New Password does not match Current Password";
}
if($current_password == $new_password) {
    $error[] = "New Password must be different than Current Password";
}
if($current_password == $confirm_new_password) {
    $error[] = "Confirm New Password must be different than Current Password";
}
if($current_password == '') {
    $error[] = "Current Password must contain at least one character";
}
if($new_password == '') {
    $error[] = "New Password must contain at least one character";
}
if($confirm_new_password == '') {
    $error[] = "Confirm New Password must contain at least one character";
}
if($current_password != $new_password) {
    $error[] = "New Password does not match Current Password";
}
if($current_password == $new_password) {
    $error[] = "New Password must be different than Current Password";
}
if($current_password == $confirm_new_password) {
    $error[] = "Confirm New Password must be different than Current Password";
}
```

php code 2



Home Profile Create Role List
Roles Assign Roles Logout

Invalid email address
Username must only contain A-Z
characters a-z, 0-9, or -

Email

Username

Password Reset

Current Password

New Password

Confirm Password

CR IE | Elements Reorder Console 80

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<div>

<div class="container" id="flash"></div>

<form method="POST" onsubmit="return valid();">

<script></script>

<script></script>

</form>

</body>

</html>

PHP profile email and user validation

The chosen email is not available.
Email test@gmail.com
Username test
Password Reset
Current Password
New Password
Confirm Password
Update Profile

```
<!---- include css and js files ---->
<html>
  <head></head>
  <body> == 30
    <nav></nav>
    <div class="container" id="flash"></div>
    <form method="POST" onsubmit="return vali...
      <script></script>
      <script></script>
    </form>
  </body>
</html>
```

email taken

The chosen username is not available.
Email test@gmail.com
Username test
Password Reset
Current Password
New Password
Confirm Password
Update Profile

```
<!---- include css and js files ---->
<html>
  <head></head>
  <body> == 30
    <nav></nav>
    <div class="container" id="flash"></div>
    <form method="POST" onsubmit="return vali...
      <script></script>
      <script></script>
    </form>
  </body>
</html>
```

username taken

Profile saved.
Current password is invalid
Password and Confirm password must match
Email test@gmail.com
Username test
Password Reset
Current Password *****

```
<!---- include css and js files ---->
<html>
  <head></head>
  <body> == 30
    <nav></nav>
    <div class="container" id="flash"></div>
    <form method="POST" onsubmit="return vali...
      <script></script>
      <script></script>
    </form>
  </body>
</html>
```

100%

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

Include the related pull request link for this feature

URL #1

https://github.com/juicejoose/jjc88-it202010/Milestone1/public_html/project/profile.php



URL

<https://github.com/juicejoose/jjc88-it2>



URL #2

<https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com/project/profile.php>



URL

<https://jjc88-it202-010-prod-b44e1de4>



URL #3

<https://github.com/juicejoose/jjc88-it20201012>



URL

<https://github.com/juicejoose/jjc88-it2>



Saved: 4/15/2025 7:28:42 PM

100%

Section #6: (1 pt.) Misc

100%

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