

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-milestone-1-2024/grade/rr42>

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

Submissions:

Submission Selection

1 Submission [active] 4/2/2024 10:15:30 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:

1. Make sure you're in Milestone1 with the latest changes pulled
2. Ensure Milestone1 has been deployed to heroku dev
3. Gather the requested evidence and fill in the explanations per each prompt
4. Save the submission and generate the output PDF
5. Put the output PDF into your local repository folder
6. add/commit/push it to GitHub
7. Merge Milestone1 into dev
8. Locally checkout dev and pull the changes
9. Create and merge a pull request from dev to prod to deploy Milestone1 to prod
10. 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
<input checked="" type="checkbox"/> #1	1	Heroku dev url should be present in the address bar
<input checked="" type="checkbox"/> #2	1	Should have thoughtful CSS applied
<input checked="" type="checkbox"/> #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)
<input checked="" type="checkbox"/> #4	1	Demonstrate email already in use message (message text doesn't need to be exact, but should be clear)
<input checked="" type="checkbox"/> #5	1	Demonstrate username already in use message (message text doesn't need to be exact, but should be clear)
<input checked="" type="checkbox"/> #6	1	Demonstrate user-friendly message of new account being created

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

it202-rr42-dev-fb744d733bee.herokuapp.com/Project/register.php

Login Register

Invalid email format [client]

Email bademail

Username testeruser001

Password password1234

Confirm password1234

Register

wrong email format

#1 Heroku dev url should be present in the address bar

#2 Should have thoughtful CSS applied

#3 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)

A screenshot of a web browser window displaying a registration form. The URL in the address bar is `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/register.php`. The page title is "Login Register". The registration form includes fields for Email, Username, Password, and Confirm. The "Email" field contains `goodemail100@test.net`. The "Username" field contains `badUser!`. The "Password" and "Confirm" fields both contain `password123`. A red error message at the top of the form states: "Username must only contain 3-16 characters a-z, 0-9, _, or - [client]". Below the form, a red box displays the error message "Wrong username format".

Checklist Items (1)

#3 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)

A screenshot of a web browser window displaying a registration form. The URL in the address bar is `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/register.php`. The page title is "Login Register". The registration form includes fields for Email, Username, Password, and Confirm. The "Email" field contains `goodemail300@njit.edu`. The "Username" field contains `goodusername300`. The "Password" field contains `badpass`. The "Confirm" field also contains `badpass`. A red error message at the top of the form states: "Password must be at least 8 characters long [client]".

Register

wrong password format

Checklist Items (1)

#3 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)

The screenshot shows a web browser window with the URL `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/register.php`. The page has a dark background with white text. At the top, there are two buttons: Login and Register. Below them is a red error message: **Passwords do not match [client]**. Underneath the message are four input fields: Email (`goodemail300@njit.edu`), Username (`goodusername300`), Password (`goodpass123`), and Confirm (`notmatching123`). The 'Confirm' field is highlighted in red, indicating a validation error. At the bottom of the form is a Register button.

passwords not matching

Checklist Items (1)

#3 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)

The screenshot shows a web browser window with the URL `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/register.php`. The page has a dark background with white text. At the top, there are two buttons: Login and Register. Below them is a red error message: **passwords not matching**. There are no visible input fields or other validation messages on this part of the screen.

All fields are required [client]

Email

Username

Password

Confirm

empty email field

Checklist Items (1)

#3 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)

The screenshot shows a web browser window with a registration form. The URL in the address bar is `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/register.php`. The page title is "Login Register". A red header bar at the top contains the text "All fields are required [client]". Below this, there are four input fields: "Email" (containing `goodemail00@njit.edu`), "Username" (empty), "Password" (containing `password123`), and "Confirm" (containing `password123`). Below the inputs is a "Register" button. At the bottom of the page, a red footer bar displays the error message "empty username field".

empty username field

Checklist Items (1)

#3 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)

required

A screenshot of a web browser window. The address bar shows the URL `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/register.php`. The page title is "Login Register". A red horizontal bar at the top displays the message "All fields are required [client]". Below this, there are four input fields: "Email" containing `goodemail00@njit.edu`, "Username" containing `ashuifohuifoygf`, "Password" (empty), and "Confirm" containing `password123`. A "Register" button is located below the password field. The background of the page is dark.

empty password field

Checklist Items (1)

#3 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

A screenshot of a web browser window, identical to the one above. The address bar shows the URL `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/register.php`. The page title is "Login Register". A red horizontal bar at the top displays the message "All fields are required [client]". Below this, there are four input fields: "Email" containing `goodemail00@njit.edu`, "Username" containing `ashuifohuifoygf`, "Password" (empty), and "Confirm" containing `password123`. A "Register" button is located below the password field. The background of the page is dark.

empty confirm field

Checklist Items (1)

#3 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

The screenshot shows a web browser window with the URL it202-rr42-dev-fb744d733bee.herokuapp.com/Project/register.php. The page title is "Login Register". A yellow error message bar at the top says "The chosen email is not available.". Below it, there are four input fields: "Email" (empty), "Username" (empty), "Password" (empty), and "Confirm" (empty). At the bottom is a "Register" button.

email already in use

Checklist Items (1)

#4 Demonstrate email already in use message (message text doesn't need to be exact, but should be clear)

The screenshot shows a web browser window with the URL it202-rr42-dev-fb744d733bee.herokuapp.com/Project/register.php. The page title is "Login Register". A yellow error message bar at the top says "The chosen username is not available.". Below it, there are four input fields: "Email" (empty), "Username" (empty), "Password" (empty), and "Confirm" (empty). At the bottom is a "Register" button.

Confirm

Register

username already in use

Checklist Items (1)

#5 Demonstrate username already in use message (message text doesn't need to be exact, but should be clear)

it202-rr42-dev-fb744d733bee.herokuapp.com/Project/register.php

Login Register

Successfully registered!

Email [REDACTED]

Username [REDACTED]

Password [REDACTED]

Confirm [REDACTED]

Register

account successfully created

Checklist Items (1)

#6 Demonstrate user-friendly message of new account being created

Task #2 - Points: 1

Text: Screenshot of the form code

Details:

Should have appropriate input types for the field

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```

register.php 1. M X

public_html > Project > register.php > script > validate
    You, 1 second ago | 2 authors (You and others)
1  <?php //rr42 4/3/2024
2  require(__DIR__ . "/../../partials/nav.php");
3  reset_session();
4  ?
5  <form onsubmit="return validate(this)" method="POST">
6      <div>
7          <label for="email">Email</label>
8          <input type="email" name="email" required />
9      </div>
10     <div>
11         <label for="username">Username</label>
12         <input type="text" name="username" required maxlength="30" />
13     </div>
14     <div>
15         <label for="pw">Password</label>
16         <input type="password" id="pw" name="password" required minlength="8" />
17     </div>
18     <div>
19         <label for="confirm">Confirm</label>
20         <input type="password" name="confirm" required minlength="8" />
21     </div>
22     <input type="submit" value="Register" />
23 </form>

```

register.php form

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
<input checked="" type="checkbox"/> #1	1	Show the JavaScript validations (include any extra files related)
<input checked="" type="checkbox"/> #2	1	Show the PHP validations (include any lib content)
<input checked="" type="checkbox"/> #3	1	Include ucid/date code comments in all screenshots (one per screenshot is sufficient)

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```

register.php 1.0 M X
public_html > Project > register.php > script > isValidEmail
24
25   <script>
26     function validateForm() { //vvvv 4/3/2024
27       var email = form.email.value.trim();
28       var username = form.username.value.trim();
29       var password = form.password.value.trim();
30       var confirm = form.confirm.value.trim();
31
32       if (email === "" || username === "" || password === "" || confirm === "") {
33         flash("All fields are required. [client]", "warning");
34         return false;
35       }
36
37       if (!isValidEmail(email)) {
38         flash("Invalid email address. [client]", "warning");
39         return false;
40       }
41
42       if (!isValidUsername(username)) {
43         flash("Username must only contain 3-16 characters a-z, 0-9, _, or -. [client]", "warning");
44         return false;
45       }
46
47       if (!isValidPassword(password)) {
48         flash("Password must be at least 8 characters long. [client]", "warning");
49         return false;
50       }
51
52       if (password !== confirm) {
53         flash("Passwords do not match. [client]", "warning");
54         return false;
55       }
56
57       return true;
58     }
59
60     function isValidEmail(email) {
61       var regex = /^[^\s@]+@[^\s@]+\.\w+$/;
62       return regex.test(email);] -- You, 13 minutes ago * updated 3s validation
63     }
64
65     function isValidUsername(username) {
66       var regex = /^[a-zA-Z0-9_-]{3,16}$/;
67       return regex.test(username);
68     }
69
70     function isValidPassword(password) {
71       return password.length >= 8;
72     }
73
74   </script>
75

```

register.php js validation

Checklist Items (1)

#1 Show the JavaScript validations (include any extra files related)

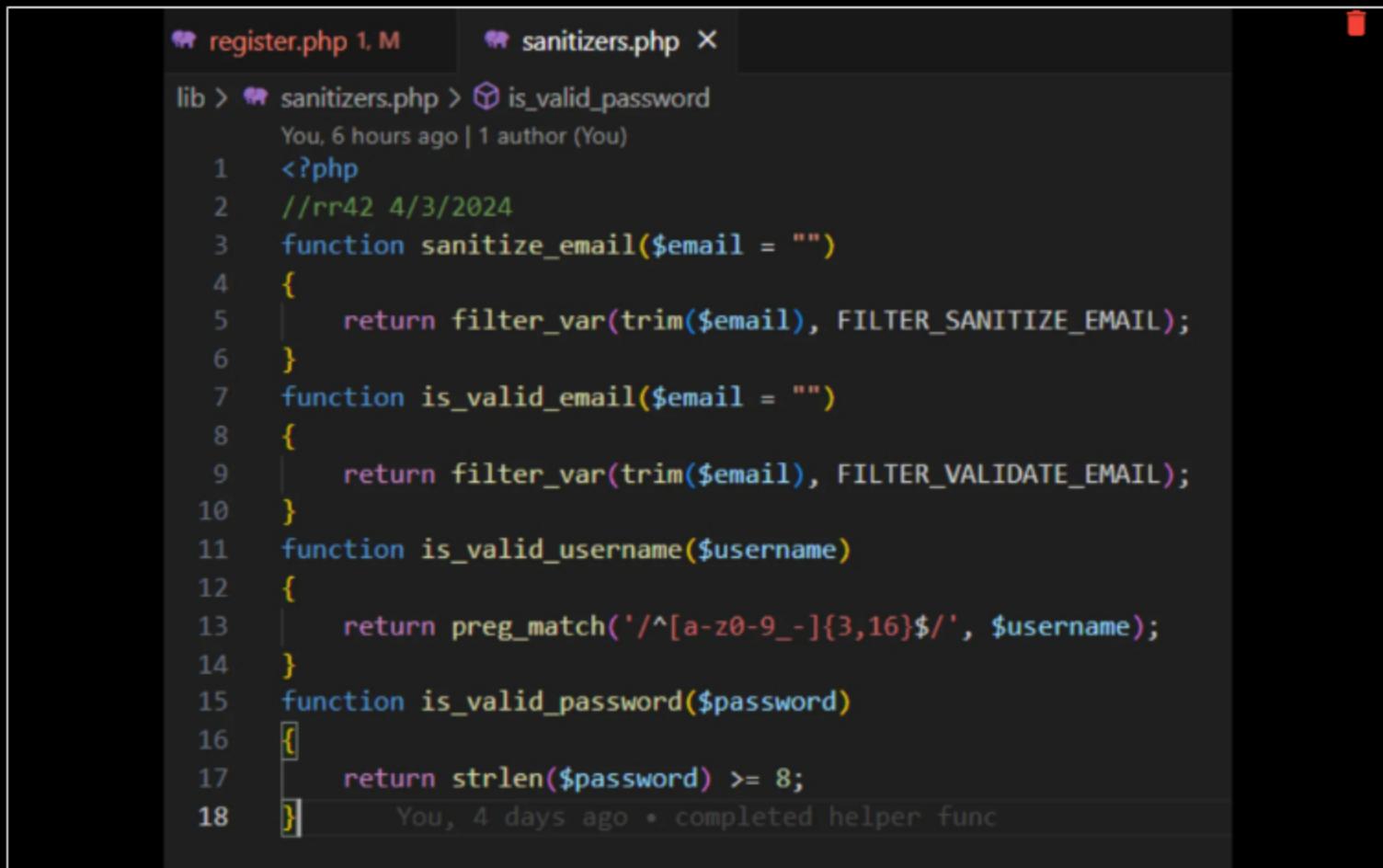
```

register.php 1.0 M X
public_html > Project > register.php >-
77  //vvvv 2; add PHP Code vvvv 4/3/2024 -- You, 3 seconds ago * uncommitted changes
78  if (!isset($_POST["email"]) || !isset($_POST["password"]) || !isset($_POST["confirm"]) || !isset($_POST["username"])) {
79    $email = $_POST["email"] ?? "", false;
80    $password = $_POST["password"] ?? "", false;
81    $confirm = $_POST["confirm"] ?? "", false;
82    $username = $_POST["username"] ?? "", false;
83    //vvvv 3
84    $hasError = false;
85    if (empty($email)) {
86      flash("Email must not be empty", "danger");
87      $hasError = true;
88    }
89    //sanitize
90    $email = sanitize_email($email);
91    //validate
92    if (!is_valid_email($email)) {
93      flash("Invalid email address", "danger");
94      $hasError = true;
95    }
96    if (!is_valid_username($username)) {
97      flash("Username must only contain 3-16 characters a-z, 0-9, _, or -. [danger]");
98      $hasError = true;
99    }
100   if (empty($password)) {
101     flash("Password must not be empty", "danger");
102     $hasError = true;
103   }
104   if (empty($confirm)) {
105     flash("Confirm password must not be empty", "danger");
106     $hasError = true;
107   }
108   if (!is_valid_password($password)) {
109     flash("Password too short", "danger");
110     $hasError = true;
111   }
112   if (
113     strlen($password) > 0 && $password !== $confirm
114   ) {
115     flash("Passwords must match", "danger");
116     $hasError = true;
117   }
118   if (!$hasError) {
119    //vvvv 4
120    $hash = password_hash($password, PASSWORD_BCRYPT);
121    $db = getDB();
122    $stmt = $db->prepare("INSERT INTO Users (email, password, username) VALUES (:email, :password, :username)");
123    try {
124      $stmt->execute([":email" => $email, ":password" => $hash, ":username" => $username]);
125      flash("Successfully registered!", "success");
126    } catch (Exception $e) {
127      users_check_duplicate($e->errorInfo);
128    }
129  }
130

```

Checklist Items (1)

#2 Show the PHP validations (include any lib content)



```

register.php 1, M sanitizers.php X
lib > sanitizers.php > is_valid_password
You, 6 hours ago | 1 author (You)
1 <?php
2 //rr42 4/3/2024
3 function sanitize_email($email = "") {
4 {
5     return filter_var(trim($email), FILTER_SANITIZE_EMAIL);
6 }
7 function is_valid_email($email = "") {
8 {
9     return filter_var(trim($email), FILTER_VALIDATE_EMAIL);
10 }
11 function is_valid_username($username) {
12 {
13     return preg_match('/^[\w\-\_]{3,16}$/', $username);
14 }
15 function is_valid_password($password) {
16 {
17     return strlen($password) >= 8;
18 }

```

You, 4 days ago • completed helper func

sanitizers.php

Checklist Items (0)

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
<input checked="" type="checkbox"/> #1	1	Password should be hashed
<input checked="" type="checkbox"/> #2	1	Should have email, password, username (unique), created, modified, and id fields
<input checked="" type="checkbox"/> #3	1	Ensure left panel or database name is present (should contain your ucid)

Task Screenshots:

Gallery Style: Large View

Small Medium Large

Properties Data Process

SELECT * FROM "Users" LIMIT 100

Search results

	id	email	password	created	modified	username
#	Q	•	•	•	•	•
> 1	1	r42@njit.edu	\$2y\$10\$hyYQOEWnclMaPn	2024-04-01 22:26:31	2024-04-01 22:26:31	r42
> 2	2	test1@test.com	\$2y\$10\$xlNpbOSZ2IlgGTw	2024-04-01 22:27:56	2024-04-01 22:27:56	test1
> 3	3	test2@test.gov	\$2y\$10\$NDuRLUbraMHICh	2024-04-01 22:43:46	2024-04-01 22:43:46	test02
> 4	8	test10@test.net	\$2y\$10\$R1M3HHydJREQ3i	2024-04-03 03:36:52	2024-04-03 03:36:52	test010
> 5	9	test01@njit.edu	\$2y\$10\$rl996gfyQcGW0oC	2024-04-03 03:48:30	2024-04-03 03:48:30	test01
> 6	10	goodemail2@njit.edu	\$2y\$10\$A3pTDOIG.okiBDi	2024-04-03 19:27:48	2024-04-03 19:27:48	testeruser011
> 7	13	testuseremail@test.net	\$2y\$10\$foV9FZQAWSCAV2	2024-04-03 19:41:04	2024-04-03 19:41:04	testuser00

Users table

Checklist Items (3)

#1 Password should be hashed

#2 Should have email, password, username (unique), created, modified, and id fields

#3 Ensure left panel or database name is present (should contain your ucid)

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:

Missing Response

Task #6 - Points: 1

Text: Include pull request links related to this feature

● Details:

Should end in /pull/#

URL #1

<https://github.com/rRahmanM6/rr42-it202-008/pull/23>



User Login (2 pts.)

▲ COLLAPSE ▲



Task #1 - Points: 1

▲ COLLAPSE ▲

Text: Screenshot of form on website page

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large



The screenshot shows a web browser window with the following details:

- Address Bar:** it202-n42-dev-fb744d733bee.herokuapp.com/Project/login.php
- Form Fields:** Email/Username (placeholder: "Email/Username"), Password (placeholder: "Password").
- Error Message:** A red horizontal bar displays the text "Invalid username".
- Buttons:** A "Login" button at the bottom left.

invalid username

Checklist Items (5)

#1 Heroku dev url should be present in the address bar

#2 Should have thoughtful CSS applied

#3 Include correct data where username is used to login

#4 Include correct data where email is used to login

#6 Demonstrate JavaScript validation for each field [can be combined] (username format, email format, password format, and each field being required)

The screenshot shows a web browser window with the URL `it202-n42-dev-fb744d733bee.herokuapp.com/Project/login.php`. The page has two buttons at the top: Login and Register. Below them is an error message: **Invalid email address**. There are two input fields: **Email/Username** and **Password**, both of which are currently empty. At the bottom left is a **Login** button.

invalid email

Checklist Items (1)

#6 Demonstrate JavaScript validation for each field [can be combined] (username format, email format, password format, and each field being required)

The screenshot shows a web browser window with the URL `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/login.php`. The page displays a login form with two buttons: "Login" and "Register". A red error message at the top states: "Password must be at least 8 characters long
". Below the message are two input fields: "Email/Username" and "Password", both of which are currently empty. The "Login" button is highlighted with a blue glow.

wrong password format

Checklist Items (1)

#6 Demonstrate JavaScript validation for each field [can be combined] (username format, email format, password format, and each field being required)

The screenshot shows a web browser window with the URL `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/login.php`. The page displays a login form with two buttons: "Login" and "Register". A red error message at the top states: "Email must be provided [client]". Below the message are two input fields: "Email/Username" and "Password", both of which are currently empty. The "Login" button is highlighted with a blue glow.

empty email field

Checklist Items (1)

#6 Demonstrate JavaScript validation for each field [can be combined] (username format, email format, password format, and each field being required)

The screenshot shows a web browser window with the URL `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/login.php`. The page title is "Login Register". There are two error messages displayed in red boxes: "Password must be provided
" and "Password must be at least 8 characters long
". Below these messages are input fields for "Email/Username" and "Password", both of which are currently empty. A "Login" button is visible below the password field.

empty password field

Checklist Items (1)

#6 Demonstrate JavaScript validation for each field [can be combined] (username format, email format, password format, and each field being required)

The screenshot shows a web browser window with the URL `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/home.php`. The page title is "Home Profile Logout". At the top, there is a red bar containing the text "Welcome, test". Below the bar, the word "Home" is prominently displayed in large white letters.

successful login

Checklist Items (2)

#5 Show success login message

#9 Demonstrate successful login message and the destination page (i.e., home or some landing/dashboard page)

A screenshot of a web browser window. The address bar shows the URL: `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/login.php`. The page content is a login form with two buttons: "Login" and "Register". Above the form, there is a red horizontal bar containing the text "Email not found". Below the form, there is a message "account doesn't exist".

account doesn't exist

Checklist Items (1)

#7 Demonstrate user-friendly message of when an account doesn't exist

A screenshot of a web browser window. The address bar shows the URL: `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/login.php`. The page content is a login form with two buttons: "Login" and "Register". Above the form, there is a red horizontal bar containing the text "Email not found".

Login Register

Invalid password

Email/Username Password **Login**

wrong password

Checklist Items (1)

#8 Demonstrate user-friendly message of when password doesn't match what's in the DB

Task #2 - Points: 1**Text: Screenshot of the form code****Checklist**

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #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)
<input type="checkbox"/> #2	1	Show JavaScript validations (include any extra files related)
<input type="checkbox"/> #3	1	Show PHP validations (include any lib content)
<input type="checkbox"/> #4	1	Include ucid/date code comments in all screenshots (one per screenshot is sufficient)

Task Screenshots:**Gallery Style: Large View****Small Medium Large****login.php M X**

public_html > Project > login.php > script > validate

You, 1 second ago | 2 authors (You and others)

```

1 <?php //rr42 4/3/2024
2 require_once(__DIR__ . "/../../partials/nav.php");
3 ?>
4 <form onsubmit="return validate(this)" method="POST">
5     <div>
6         <label for="email">Email/Username</label>
7         <input type="text" name="email" required />
8     </div>
9     <div>
10        <label for="pw">Password</label>
11        <input type="password" id="pw" name="password" required minlength="8" />
12    </div>
13    <input type="submit" value="Login" />
14 </form>

```

login.php form (combined username/email field)

Checklist Items (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)

```

login.php M flash_messages.php M X sanitizers.php

lib > flash_messages.php > getMessages
You, 5 seconds ago | 1 author (You)
1 <?php //rr42 4/3/2024
2 function flash($msg = "", $color = "info")
3 {
4     $message = ["text" => $msg, "color" => $color];
5     if (isset($_SESSION['flash'])) {
6         array_push($_SESSION['flash'], $message);
7     } else {
8         $_SESSION['flash'] = array();
9         array_push($_SESSION['flash'], $message);
10    }
11 }
12
13 function getMessages()
14 {
15     if (isset($_SESSION['flash'])) {
16         $flashes = $_SESSION['flash'];
17         $_SESSION['flash'] = array();
18         return $flashes;
19     }
20     return array();
21 }

```

flash_messages.php

Checklist Items (1)

#3 Show PHP validations (include any lib content)

login.php M

flash_messages.php M

sanitizers.php X

lib > sanitizers.php > is_valid_password

You, 45 minutes ago | 1 author (You)

```
1 <?php
2 //rr42 4/3/2024
3 function sanitize_email($email = "") {
4     return filter_var(trim($email), FILTER_SANITIZE_EMAIL);
5 }
6 function is_valid_email($email = "") {
7     return filter_var(trim($email), FILTER_VALIDATE_EMAIL);
8 }
9 function is_valid_username($username) {
10    return preg_match('/^[\w\0-9_-]{3,16}$/', $username);
11 }
12 function is_valid_password($password) {
13    return strlen($password) >= 8;
14 }
```

You, 4 days ago • completed helper func

sanitizers.php

Checklist Items (1)

#3 Show PHP validations (include any lib content)

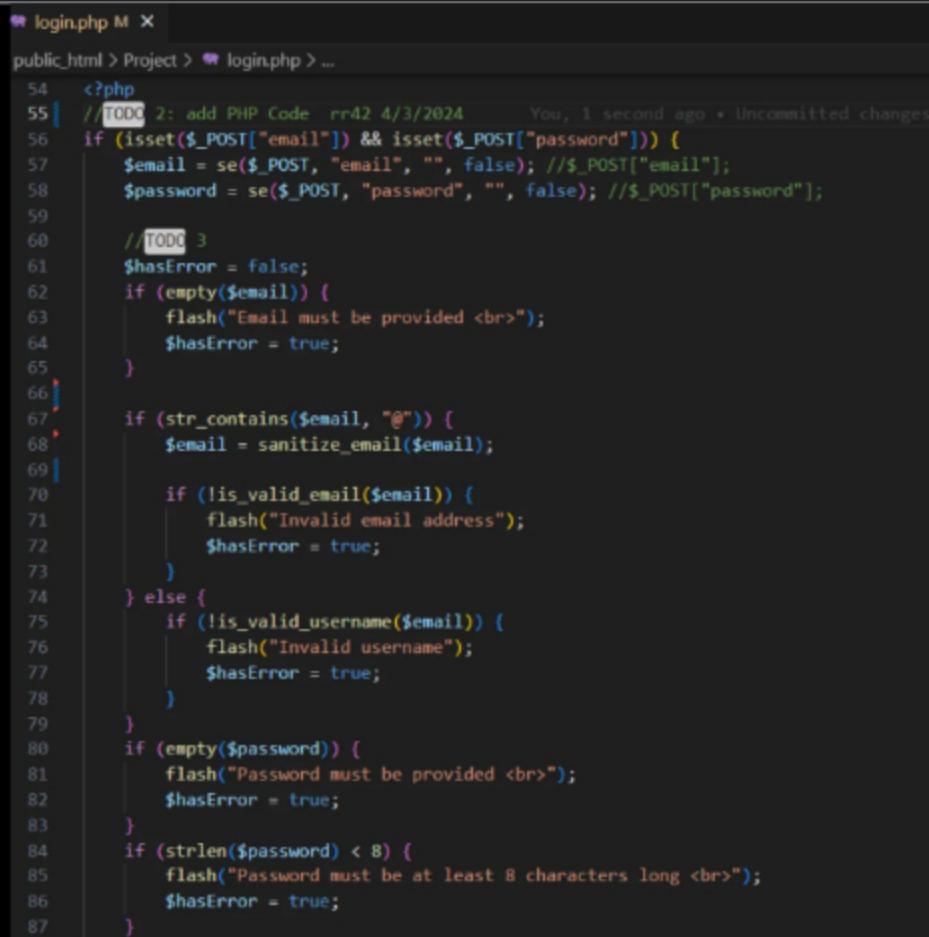
login.php M

public_html > Project > login.php > form

```
16 <script> //rr42 4/3/2024
17 function validate(form) {
18     var emailField = form.email.value.trim();
19     var passwordField = form.password.value.trim();
20
21     if (emailField === "") {
22         flash("Email/Username must be provided [client]", "warning");
23         return false;
24     }
25
26     if (passwordField === "") {
27         flash("Password must be provided [client]", "warning");
28         return false;
29     }
30
31     if (emailField.includes("@")) {
32         if (!isValidEmail(emailField)) {
33             flash("Please enter a valid email address [client]", "warning");
34             return false;
35         }
36     }
37
38     if (passwordField.length < 8) {
39         flash("Password must be at least 8 characters long [client]", "warning");
40         return false;
41     }
42
43     return true;
44 }
45
46 function isValidEmail(email) {
47     var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
48     return emailRegex.test(email);
49 }
50
51 </script>
```

Checklist Items (1)

#2 Show JavaScript validations (include any extra files related)



```

54  <?php
55  // TODO 2: add PHP Code rr42 4/3/2024      You, 1 second ago + Uncommitted changes
56  if (isset($_POST["email"]) && isset($_POST["password"])) {
57      $email = se($_POST, "email", "", false); //$_POST["email"];
58      $password = se($_POST, "password", "", false); //$_POST["password"];
59
60      // TODO 3
61      $hasError = false;
62      if (empty($email)) {
63          flash("Email must be provided <br>");
64          $hasError = true;
65      }
66
67      if (str_contains($email, "@")) {
68          $email = sanitize_email($email);
69
70          if (!is_valid_email($email)) {
71              flash("Invalid email address");
72              $hasError = true;
73          }
74      } else {
75          if (!is_valid_username($email)) {
76              flash("Invalid username");
77              $hasError = true;
78          }
79      }
80      if (empty($password)) {
81          flash("Password must be provided <br>");
82          $hasError = true;
83      }
84      if (strlen($password) < 8) {
85          flash("Password must be at least 8 characters long <br>");
86          $hasError = true;
87      }

```

login.php php validations

Checklist Items (1)

#3 Show PHP validations (include any lib content)

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
<input type="checkbox"/> #1	1	Don't just show code, translate things to plain English
<input type="checkbox"/> #2	1	Explain how the session works and why/how it's used

Response:

Missing Response

Task #4 - Points: 1

[COLLAPSE](#)

TASK #4 - Points: 1

Text: Include pull request links related to this feature

Details:

Should end in /pull/#

URL #1

<https://github.com/rRahmanM6/rr42-it202-008/pull/27>

URL #2

<https://github.com/rRahmanM6/rr42-it202-008/pull/42>



User Logout (1 pt.)

[COLLAPSE](#)



Task #1 - Points: 1

Text: Capture the following screenshots

Checklist

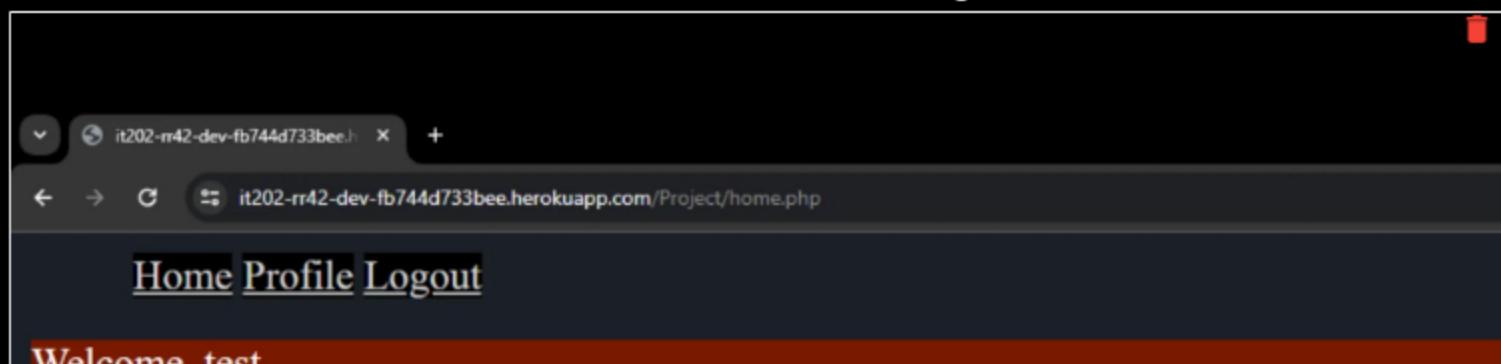
*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Screenshot of the navigation when logged in (site)
<input type="checkbox"/> #2	1	Screenshot of the redirect to login with the user-friendly logged-out message (site)
<input type="checkbox"/> #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](#)

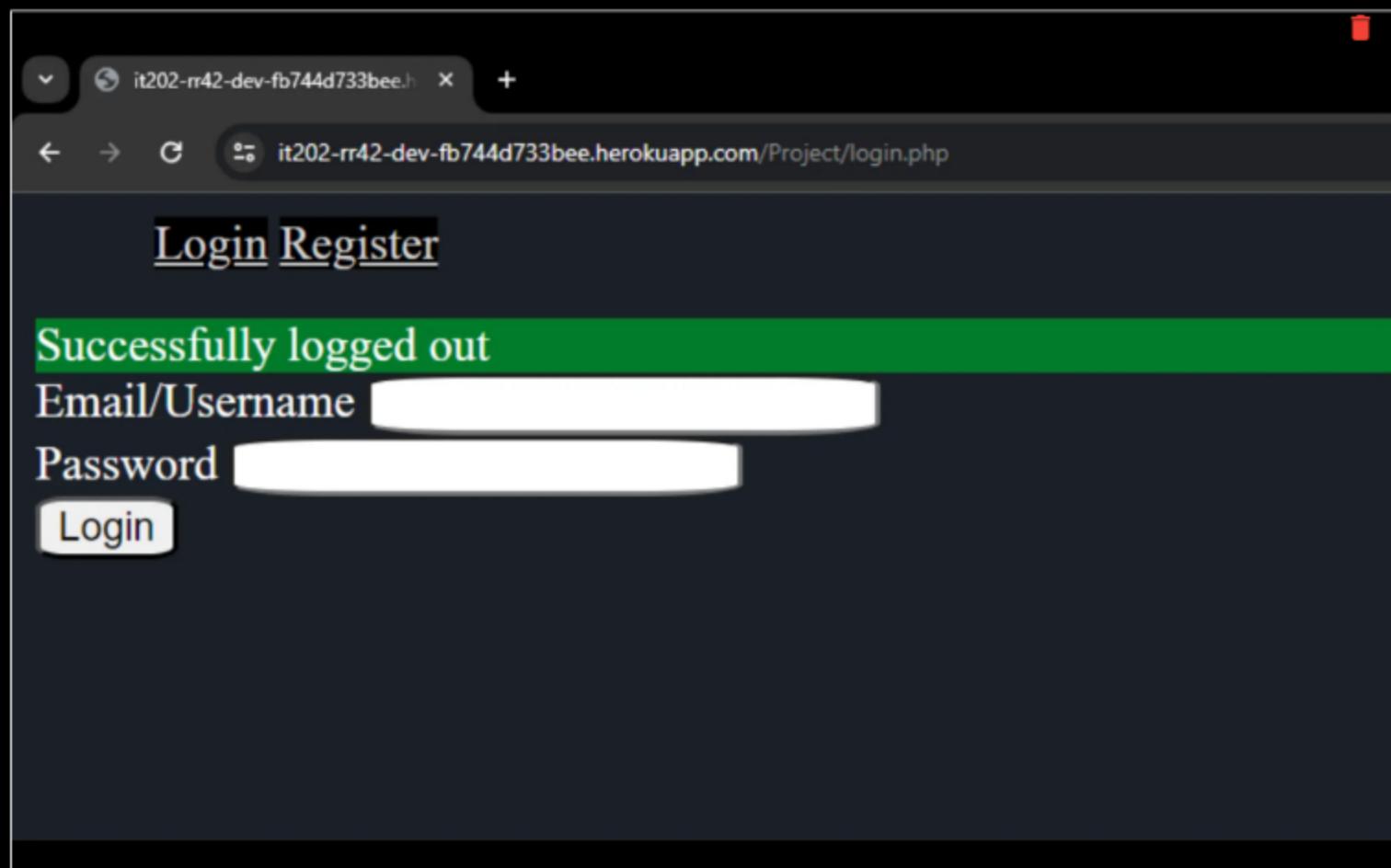


The screenshot shows a web browser window with the URL it202-rr42-dev-fb744d733bee.herokuapp.com/. The page displays a navigation bar with three links: "Home", "Profile", and "Logout". Below the navigation bar, there is a red banner with the text "Welcome, test". The main content area of the page is blank.

logged in

Checklist Items (1)

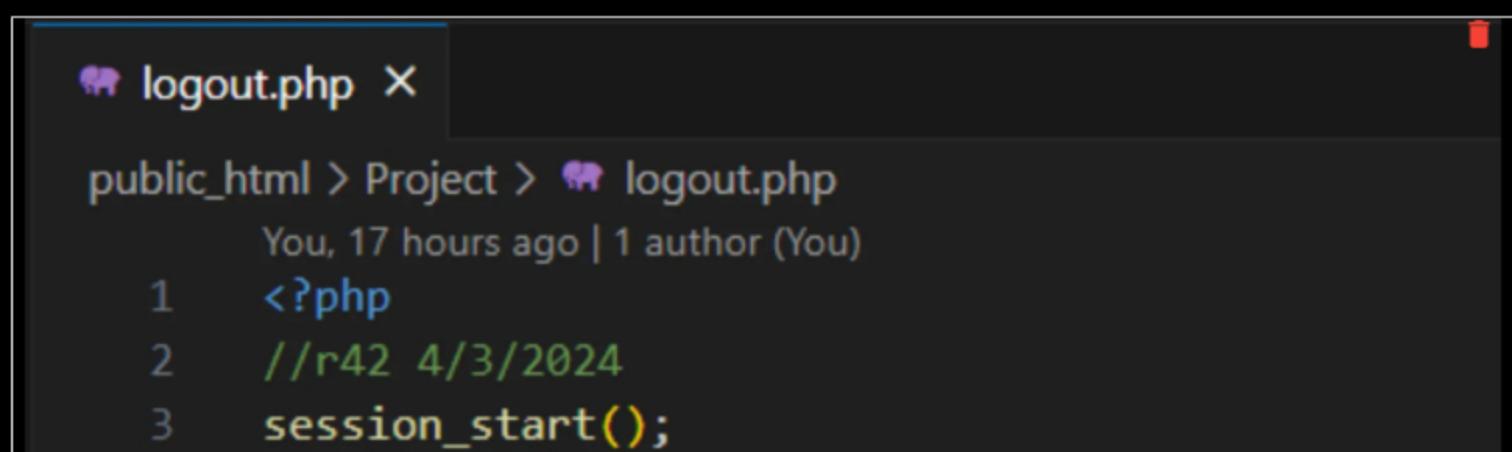
#1 Screenshot of the navigation when logged in (site)



logged out

Checklist Items (1)

#2 Screenshot of the redirect to login with the user-friendly logged-out message (site)



```
4 require(__DIR__ . "/../../lib/functions.php");
5 reset_session();
6
7 flash("Successfully logged out", "success");
8 header("Location: login.php"); You, 4 days
```

logout.php

Checklist Items (1)

#3 Screenshot of the logout-related code showing the session is destroyed (code). Ensure ucid/date comment is present.

Task #2 - Points: 1

Text: Include pull request links related to this feature

ⓘ Details:

Should end in /pull/#

URL #1

<https://github.com/rRahmanM6/rr42-it202-008/pull/27>

Basic Security Rules and Roles (2 pts.)

Task #1 - Points: 1

Text: Authentication Screenshots

Checklist

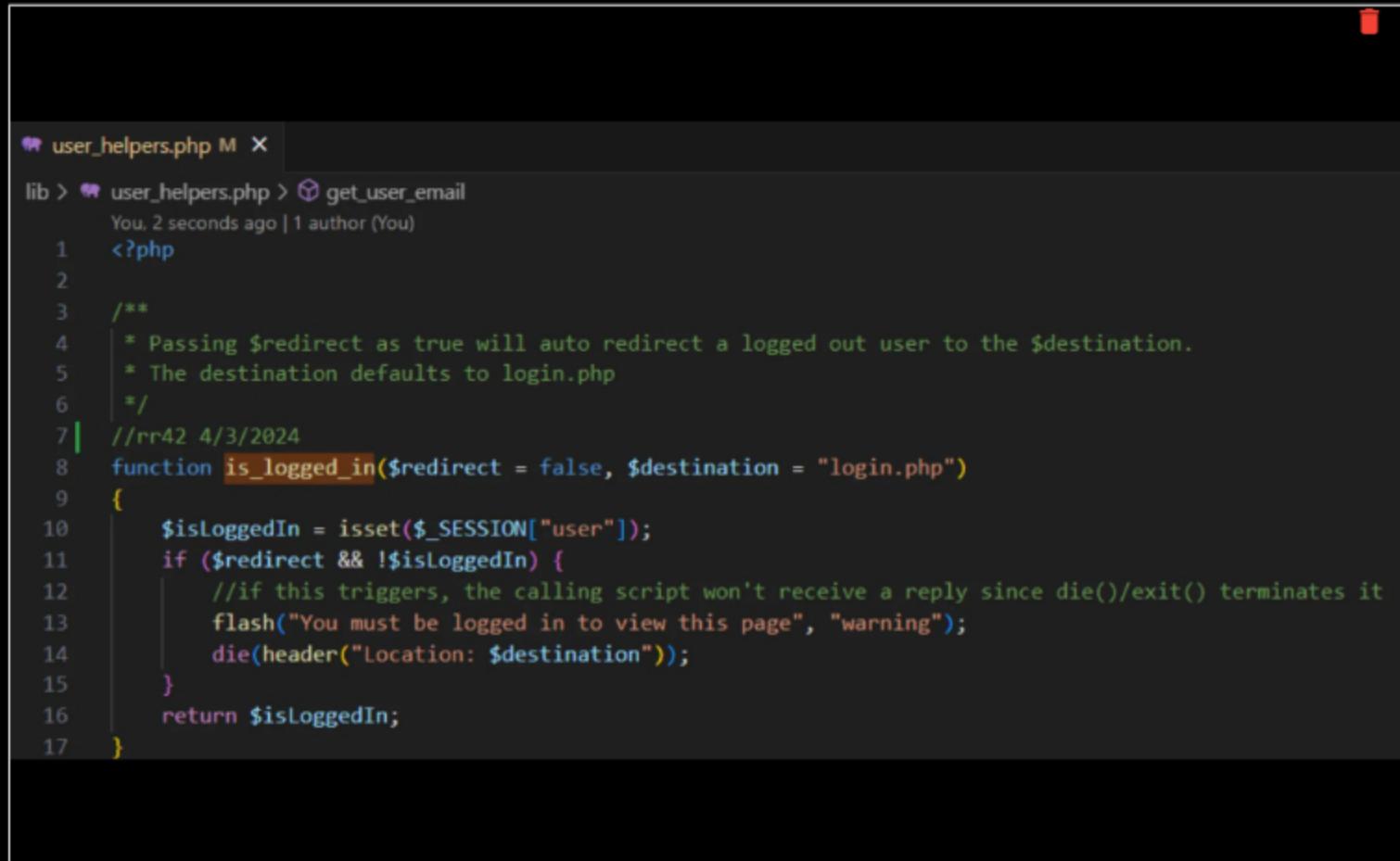
*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Screenshot of the function that checks if a user is logged in
<input checked="" type="checkbox"/> #2	1	Screenshot of the login check function being used (i.e., profile likely). Also caption what pages it's used on
<input checked="" type="checkbox"/> #3	1	Include ucid/date code comments in all screenshots (one per screenshot is sufficient)
<input checked="" type="checkbox"/> #4	1	Demonstrate the user-friendly message of trying to manually access a login-protected page while being logged out

Task Screenshots:

Gallery Style: Large View

Small Medium Large

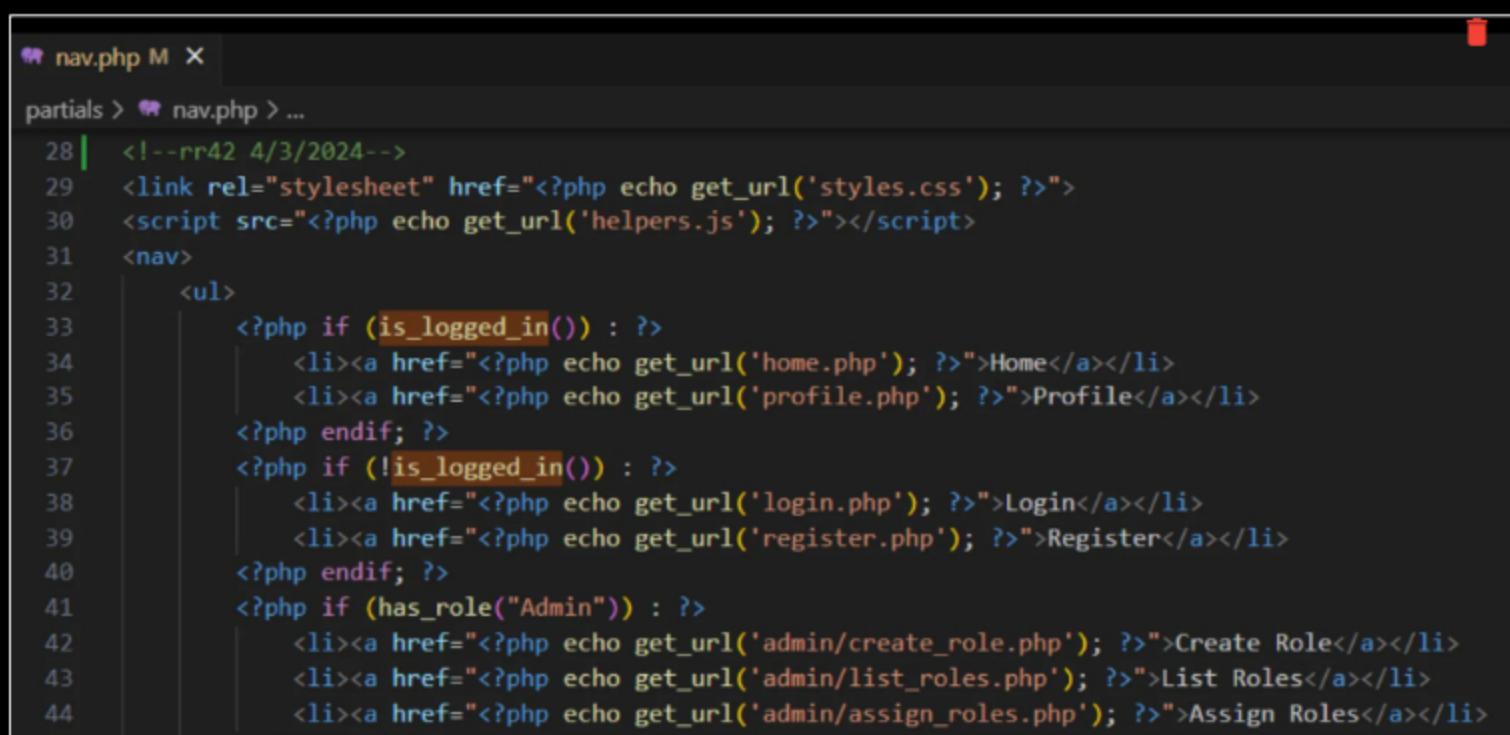


```
lib > user_helpers.php < M X
lib > user_helpers.php > get_user_email
You, 2 seconds ago | 1 author (You)
1 <?php
2
3 /**
4 * Passing $redirect as true will auto redirect a logged out user to the $destination.
5 * The destination defaults to login.php
6 */
7 //rr42 4/3/2024
8 function is_logged_in($redirect = false, $destination = "login.php")
9 {
10     $isLoggedIn = isset($_SESSION["user"]);
11     if ($redirect && !$isLoggedIn) {
12         //if this triggers, the calling script won't receive a reply since die()/exit() terminates it
13         flash("You must be logged in to view this page", "warning");
14         die(header("Location: $destination"));
15     }
16     return $isLoggedIn;
17 }
```

is_logged_in

Checklist Items (1)

#1 Screenshot of the function that checks if a user is logged in



```
partials > nav.php < M X ...
28 | <!--rr42 4/3/2024-->
29 | <link rel="stylesheet" href="php echo get_url('styles.css'); ?&gt;"&gt;
30 | &lt;script src="<?php echo get_url('helpers.js'); ?&gt;"&gt;&lt;/script&gt;
31 | &lt;nav&gt;
32 |     &lt;ul&gt;
33 |         &lt;?php if (is_logged_in()) : ?&gt;
34 |             &lt;li&gt;&lt;a href="<?php echo get_url('home.php'); ?&gt;"&gt;Home&lt;/a&gt;&lt;/li&gt;
35 |             &lt;li&gt;&lt;a href="<?php echo get_url('profile.php'); ?&gt;"&gt;Profile&lt;/a&gt;&lt;/li&gt;
36 |         &lt;?php endif; ?&gt;
37 |         &lt;?php if (!is_logged_in()) : ?&gt;
38 |             &lt;li&gt;&lt;a href="<?php echo get_url('login.php'); ?&gt;"&gt;Login&lt;/a&gt;&lt;/li&gt;
39 |             &lt;li&gt;&lt;a href="<?php echo get_url('register.php'); ?&gt;"&gt;Register&lt;/a&gt;&lt;/li&gt;
40 |         &lt;?php endif; ?&gt;
41 |         &lt;?php if (has_role("Admin")) : ?&gt;
42 |             &lt;li&gt;&lt;a href="<?php echo get_url('admin/create_role.php'); ?&gt;"&gt;Create Role&lt;/a&gt;&lt;/li&gt;
43 |             &lt;li&gt;&lt;a href="<?php echo get_url('admin/list_roles.php'); ?&gt;"&gt;List Roles&lt;/a&gt;&lt;/li&gt;
44 |             &lt;li&gt;&lt;a href="<?php echo get_url('admin/assign_roles.php'); ?&gt;"&gt;Assign Roles&lt;/a&gt;&lt;/li&gt;</pre
```

```

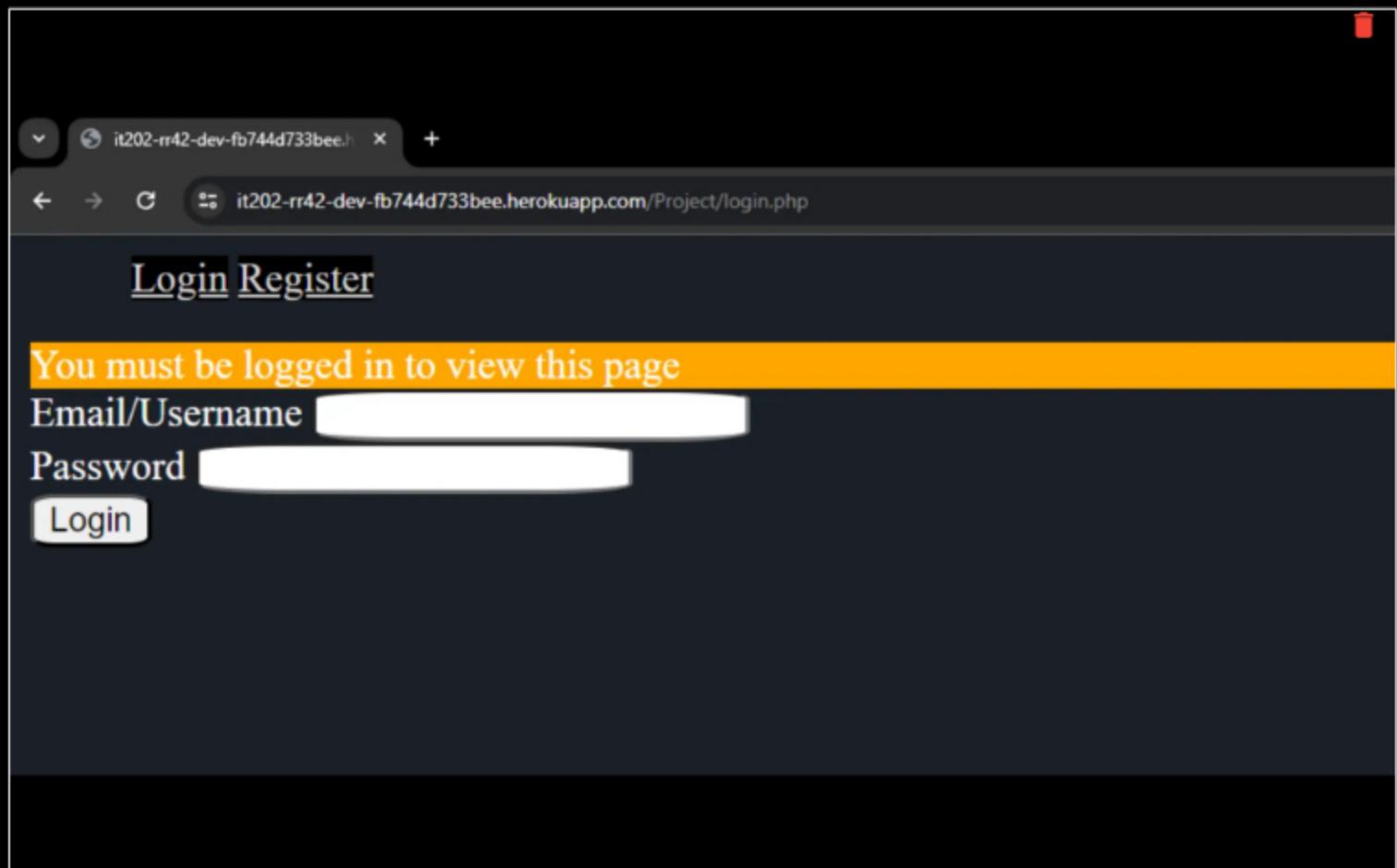
45     <?php endif; ?>
46     <?php if (is_logged_in()) : ?>
47         |     <li><a href=<?php echo get_url('logout.php'); ?>">Logout</a></li>
48     <?php endif; ?>
49     </ul>
50 </nav>

```

is_logged_in is used in home and profile

Checklist Items (1)

#2 Screenshot of the login check function being used (i.e., profile likely). Also caption what pages it's used on



trying to access profile page while logged out

Checklist Items (1)

#4 Demonstrate the user-friendly message of trying to manually access a login-protected page while being logged out

Task #2 - Points: 1

Text: Authorization Screenshots

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Screenshot of the function that checks for a specific role
#2	1	Screenshot of the role check function being used. Also caption what pages it's used

		on
#3	1	Include ucid/date code comments in all screenshots (one per screenshot is sufficient)
#4	1	Demonstrate the user-friendly message of trying to manually access a role-protected page while being logged out

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

```
18 //rr42 4/3/2024
19 function has_role($role) {
20     if (is_logged_in() && isset($_SESSION["user"]["roles"])) {
21         foreach ($_SESSION["user"]["roles"] as $r) {
22             if ($r["name"] === $role) {
23                 return true;
24             }
25         }
26     }
27     return false;
28 }
```

has_role() from user_helpers.php

Checklist Items (1)

#1 Screenshot of the function that checks for a specific role

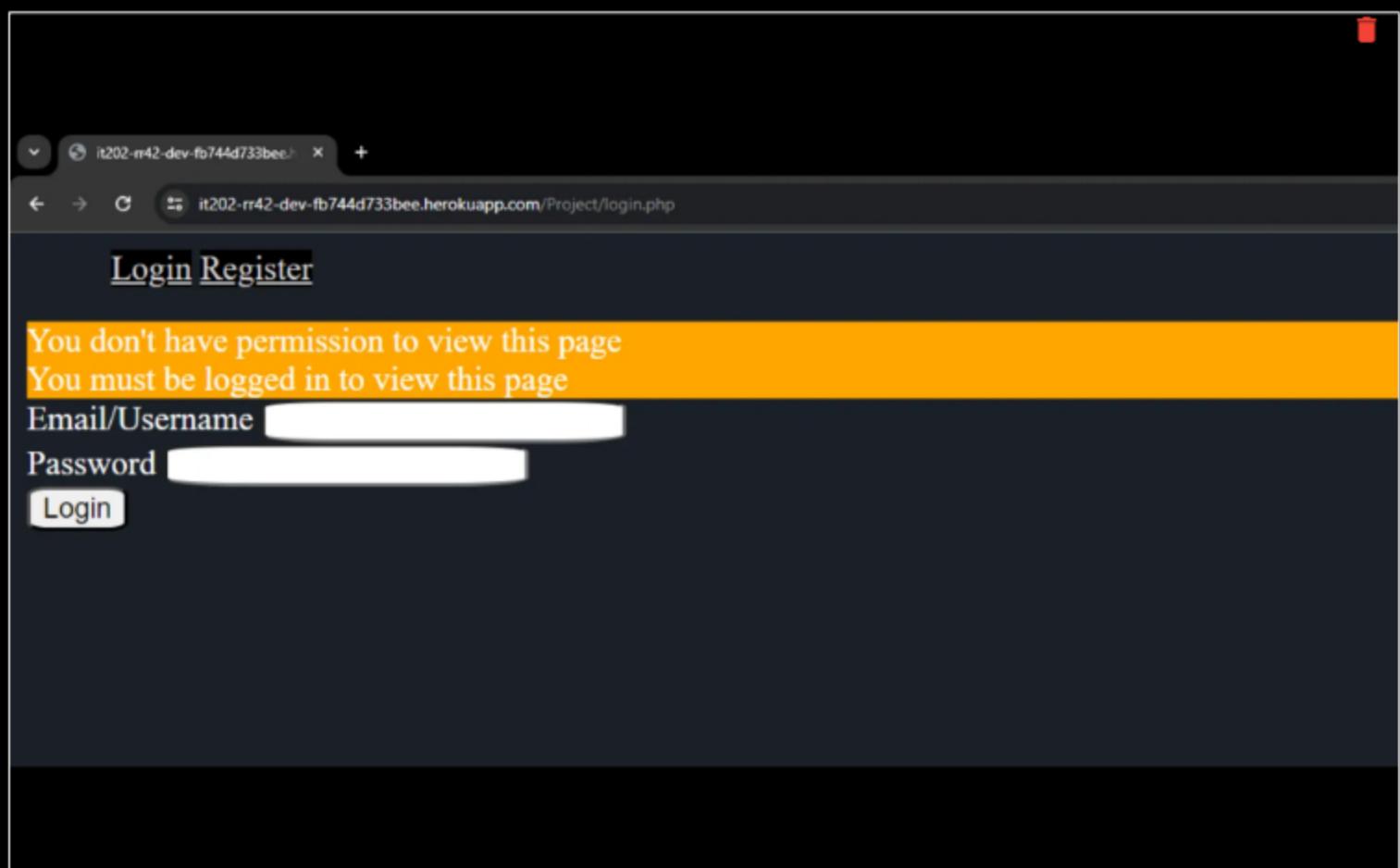
```
28 <!--rr42 4/3/2024-->
29 <link rel="stylesheet" href="=php echo get_url('styles.css'); ?&gt;"&gt;
30 &lt;script src="<?=php echo get_url('helpers.js'); ?&gt;"&gt;&lt;/script&gt;
31 &lt;nav&gt;
32 &lt;ul&gt;
33 &lt;?php if (is_logged_in()) : ?&gt;
34     &lt;li&gt;&lt;a href="<?=php echo get_url('home.php'); ?&gt;"&gt;Home&lt;/a&gt;&lt;/li&gt;
35     &lt;li&gt;&lt;a href="<?=php echo get_url('profile.php'); ?&gt;"&gt;Profile&lt;/a&gt;&lt;/li&gt;
36 &lt;?php endif; ?&gt;
37 &lt;?php if (!is_logged_in()) : ?&gt;
38     &lt;li&gt;&lt;a href="<?=php echo get_url('login.php'); ?&gt;"&gt;Login&lt;/a&gt;&lt;/li&gt;
39     &lt;li&gt;&lt;a href="<?=php echo get_url('register.php'); ?&gt;"&gt;Register&lt;/a&gt;&lt;/li&gt;
40 &lt;?php endif; ?&gt;</pre
```

```
41 <?php if (has_role("Admin")) : ?>
42     <li><a href=<?php echo get_url('admin/create_role.php'); ?>">Create Role</a></li>
43     <li><a href=<?php echo get_url('admin/list_roles.php'); ?>">List Roles</a></li>
44     <li><a href=<?php echo get_url('admin/assign_roles.php'); ?>">Assign Roles</a></li>
45 <?php endif; ?>
46 <?php if (is_logged_in()) : ?>
47     <li><a href=<?php echo get_url('logout.php'); ?>">Logout</a></li>
48 <?php endif; ?>
49 </ul>
50 </nav>
```

used in nav.php, assign_roles.php, create_role.php, and list_roles.php

Checklist Items (1)

#2 Screenshot of the role check function being used. Also caption what pages it's used on



attempting to access assign_roles page while logged out

Checklist Items (1)

#4 Demonstrate the user-friendly message of trying to manually access a role-protected page while being logged out

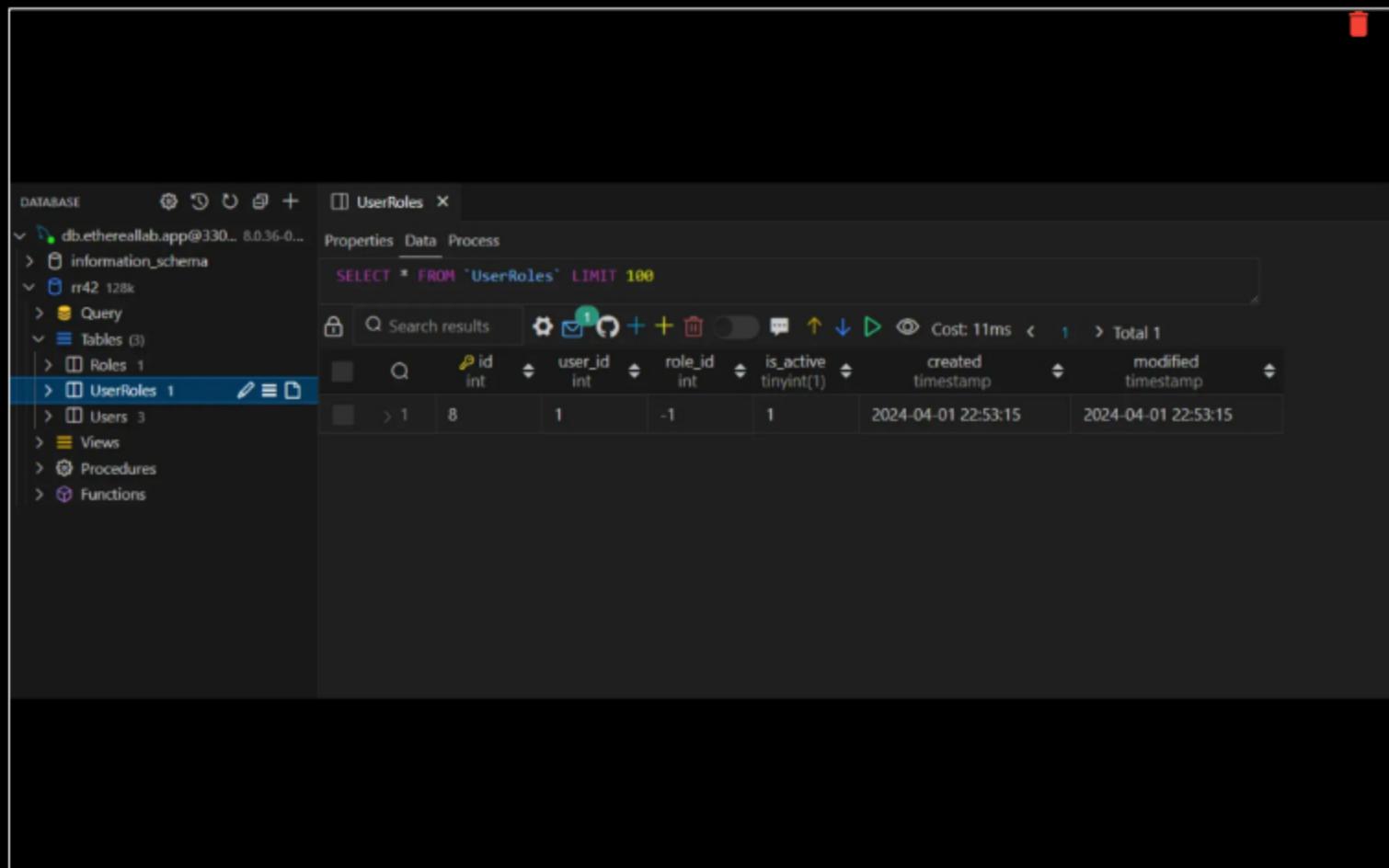
Task #3 - Points: 1

Text: Screenshots of UserRoles and Roles Tables

#	Points	Details
<input checked="" type="checkbox"/> #1	1	At least one valid and enabled User->Role reference (UserRoles table)
<input checked="" type="checkbox"/> #2	1	UserRoles Table should have id, user_id, role_id, is_active, created, and modified columns
<input checked="" type="checkbox"/> #3	1	Roles Table should have id, name, description, is_active, modified, and created columns
<input checked="" type="checkbox"/> #4	1	At least one valid and enabled Role (Roles table)
<input checked="" type="checkbox"/> #5	1	Ensure left panel or database name is present in each table screenshot (should contain your ucid)

Task Screenshots:

Gallery Style: Large View

[Small](#)[Medium](#)[Large](#)

userRoles table

Checklist Items (2)

#1 At least one valid and enabled User->Role reference (UserRoles table)

#2 UserRoles Table should have id, user_id, role_id, is_active, created, and modified columns

The screenshot shows the MySQL Workbench interface. On the left, the database tree for 'db.ethereallab.app@330...' is visible, showing 'information_schema', 'rr42 128k', 'Tables (3)', 'Roles 1', 'UserRoles 1', 'Users 3', 'Views', 'Procedures', and 'Functions'. The main area displays the 'Roles' table structure with columns: id (int), name (varchar(20)), description (varchar(100)), is_active (tinyint(1)), created (timestamp), and modified (timestamp). A query window shows the result of the SQL command 'SELECT * FROM `Roles` LIMIT 100', which returns one row: id=1, name='Admin', description='', is_active=1, created='2024-04-01 22:34:27', and modified='2024-04-01 22:34:27'.

Roles table

Checklist Items (2)

#3 Roles Table should have id, name, description, is_active, modified, and created columns

#4 At least one valid and enabled Role (Roles table)

Task #4 - Points: 1

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

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	What's the purpose of the UserRoles table?
<input type="checkbox"/> #2	1	How does Roles.is_active differ from UserRoles.is_active?

Response:

The userRoles table keeps track of any users that are assigned a specific role. If a user is assigned any role, that assignment will appear in the UserRoles table.

Roles.active specifies whether a role is available to be assigned. UserRoles.is_active specifies whether the user's assigned role is active.

Task #5 - Points: 1

Text: Include pull request links related to this feature

1 Details:

Should end in /pull/#

URL #1

<https://github.com/rRahmanM6/rr42-it202-008/pull/32>

User Profile (2 pts.)

[COLLAPSE](#)

Task #1 - Points: 1

Text: View Profile Website Page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Heroku dev url should be present in the address bar
<input checked="" type="checkbox"/> #2	1	Should have thoughtful CSS applied
<input checked="" type="checkbox"/> #3	1	Show the profile form correctly populated on page load (username, email)
<input checked="" type="checkbox"/> #4	1	Should have the following fields: username, email, current password, new password, confirm password (or similar)

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

it202-rr42-dev-fb744d733bee.herokuapp.com/Project/profile.php

Home Profile Logout

Email test@test.com

Username test

Password Reset

Current Password

New Password

Confirm Password

Update Profile

profile page

Checklist Items (4)

#1 Heroku dev url should be present in the address bar

#2 Should have thoughtful CSS applied

#3 Show the profile form correctly populated on page load (username, email)

#4 Should have the following fields: username, email, current password, new password, confirm password (or similar)

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

Details:

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

Response:

Missing Response

Task #3 - Points: 1

Text: Edit Profile Website Page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Heroku dev url should be present in the address bar
<input checked="" type="checkbox"/> #2	1	Should have thoughtful CSS applied
<input checked="" type="checkbox"/> #3	1	Demonstrate with before and after of a username change (including success message)
<input checked="" type="checkbox"/> #4	1	Demonstrate with a before and after of an email change (including success message)
<input checked="" type="checkbox"/> #5	1	Demonstrate the success message of updating password
<input checked="" type="checkbox"/> #6	1	Demonstrate JavaScript user-friendly validation messages (email format, username format, password format, new password matching confirm password)
		Demonstrate RUP user friendly validation messages (desired email is already in use)

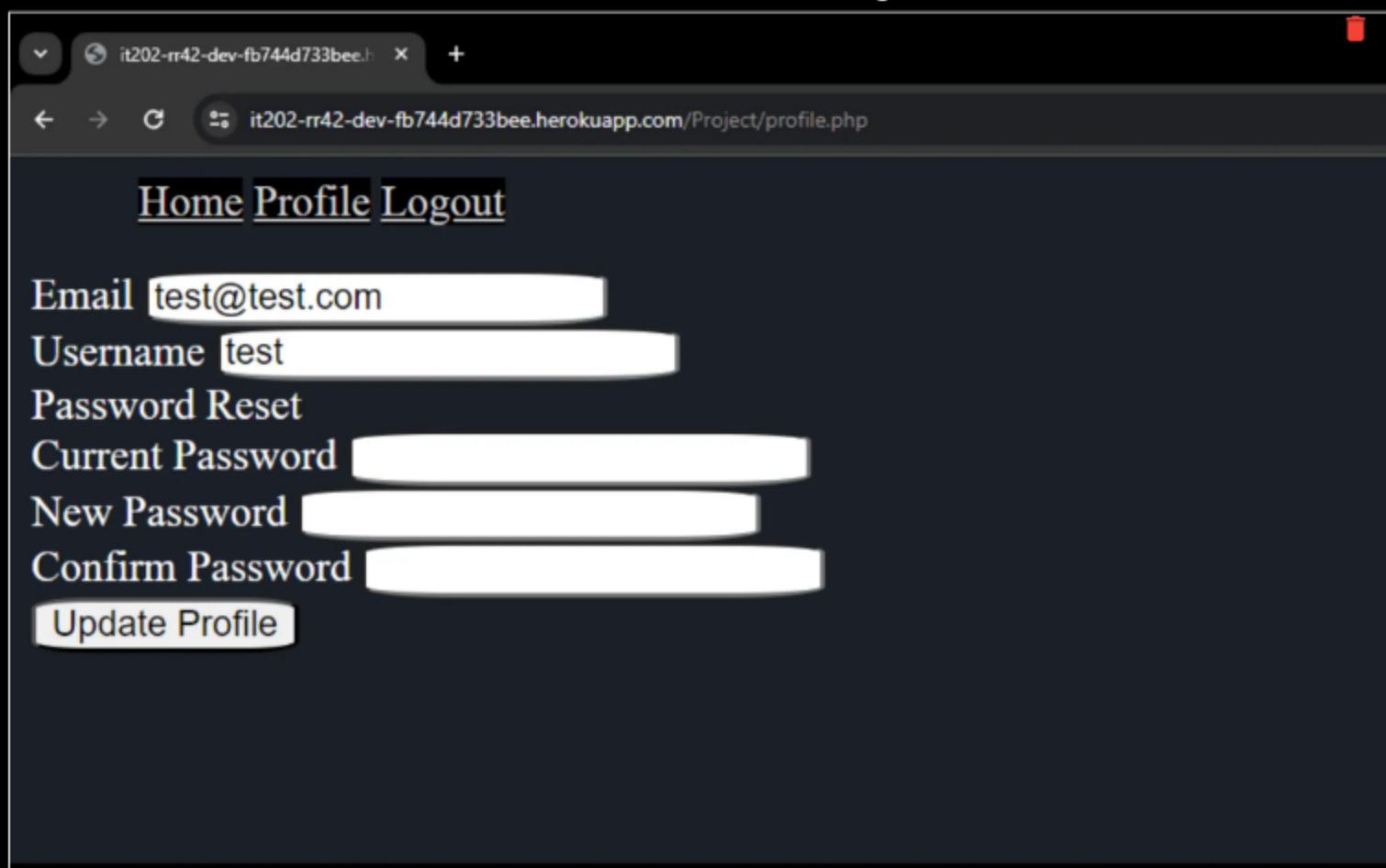
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

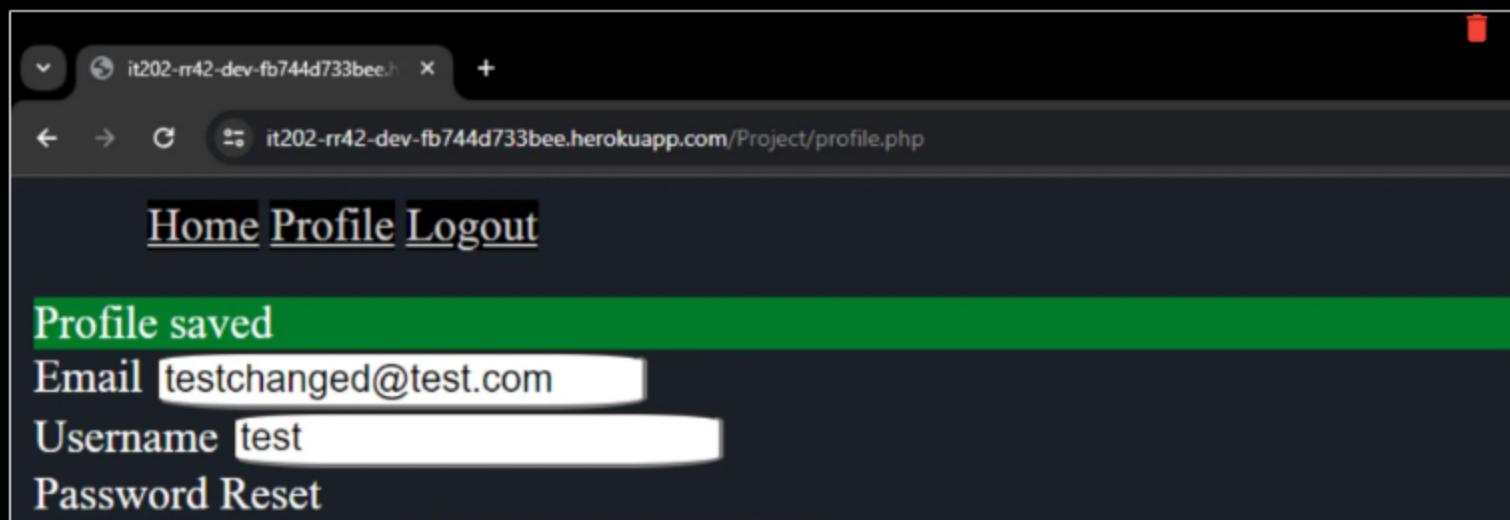


before any changes

Checklist Items (2)

#1 Heroku dev url should be present in the address bar

#2 Should have thoughtful CSS applied



Current Password [REDACTED]

New Password [REDACTED]

Confirm Password [REDACTED]

Update Profile

updated email

Checklist Items (1)

#4 Demonstrate with a before and after of an email change (including success message)

it202-rr42-dev-fb744d733bee.herokuapp.com/Project/profile.php

Home Profile Logout

Profile saved

Email testchanged@test.com

Username testchanged

Password Reset

New Password [REDACTED]

Confirm Password [REDACTED]

Update Profile

updated username

Checklist Items (1)

#3 Demonstrate with before and after of a username change (including success message)

https://it202-rr42-dev-fb744d733bee.herokuapp.com/Project/profile.php

Home Profile Logout

Profile saved

Password reset

Email testchanged@test.com

Username testchanged

Password Reset

Current Password

New Password

Confirm Password

Update Profile

password changed

Checklist Items (1)

#5 Demonstrate the success message of updating password

A screenshot of a web browser window showing a profile update form. The URL in the address bar is `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/profile.php`. The page has a dark background with white text. At the top, there are navigation links: Home, Profile, and Logout. Below these, there is an error message: "Please enter a valid email address. [client]". The "Email" field contains the value "badEmail". The "Username" field contains "tester". The "Password Reset" section is present but empty. The "Current Password" field is empty. The "New Password" field is empty. The "Confirm Password" field is empty. A "Update Profile" button is at the bottom. At the very bottom of the page, there is a footer message: "incorrect email format".

Checklist Items (1)

#6 Demonstrate JavaScript user-friendly validation messages (email format, username format, password format, new password matching confirm password)

it202-rr42-dev-fb744d733bee.h it202-rr42-dev-fb744d733bee.herokuapp.com/Project/profile.php

[Home](#) [Profile](#) [Logout](#)

Invalid username [client]

Email

Username

Password Reset

Current Password

New Password

Confirm Password

Update Profile

invalid username format

Checklist Items (1)

#6 Demonstrate JavaScript user-friendly validation messages (email format, username format, password format, new password matching confirm password)

it202-rr42-dev-fb744d733bee.h it202-rr42-dev-fb744d733bee.herokuapp.com/Project/profile.php

[Home](#) [Profile](#) [Logout](#)

Password must be at least 8 characters long. [client]

Email

Username

Password Reset

Current Password

New Password

Confirm Password

Update Profile

invalid password format

Checklist Items (1)

#6 Demonstrate JavaScript user-friendly validation messages (email format, username format, password format, new password matching confirm password)

A screenshot of a web browser window. The address bar shows the URL `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/profile.php`. The page title is "Profile". Below the title, there are several input fields and labels:

- Email: `tester@test.com`
- Username: `tester`
- Password Reset
- Current Password: `password`
- New Password: `newpass123`
- Confirm Password: `newpass1234`

Below the input fields is a button labeled "Update Profile". A yellow horizontal bar at the top of the page contains the text "Password and Confirm password must match. [client]". At the bottom of the page, a green horizontal bar contains the text "passwords not matching".

Checklist Items (1)

#6 Demonstrate JavaScript user-friendly validation messages (email format, username format, password format, new password matching confirm password)

A screenshot of a web browser window. The address bar shows the URL `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/profile.php`. The page title is "Profile". Below the title, there are several input fields and labels:

- The chosen email is not available.
- Password reset
- Email: `tester@test.com`
- Username: `tester`
- Password Reset
- Current Password: `password`
- New Password: `newpass123`
- Confirm Password: `newpass1234`

Update Profile

email already in use

Checklist Items (1)

#7 Demonstrate PHP user-friendly validation messages (desired email is already in use, desired username is already in use, current password doesn't match what's in the DB)

The screenshot shows a web browser window with the URL `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/profile.php`. The page has a dark background with some UI elements like a red trash icon in the top right corner.

The form fields are as follows:

- Email: `tester@test.com`
- Username: `tester`
- Password Reset (disabled button)
- Current Password (disabled button)
- New Password (disabled button)
- Confirm Password (disabled button)
- Update Profile (button)

Validation messages are displayed in colored bars:

- An orange bar at the top says "The chosen username is not available."
- A green bar below it says "Password reset".
- A yellow bar below that says "username already in use".

Checklist Items (1)

#7 Demonstrate PHP user-friendly validation messages (desired email is already in use, desired username is already in use, current password doesn't match what's in the DB)

The screenshot shows a web browser window with the URL `it202-rr42-dev-fb744d733bee.herokuapp.com/Project/profile.php`. The page has a dark background with some UI elements like a red trash icon in the top right corner.

The form fields are as follows:

- Email: `tester@test.com`
- Username: `tester`
- Profile saved (green bar)
- Current password is invalid (yellow bar)
- Email: `tester@test.com`
- Username: `tester`

Password Reset

Current Password [REDACTED]

New Password [REDACTED]

Confirm Password [REDACTED]

Update Profile

wrong password

Checklist Items (1)

#7 Demonstrate PHP user-friendly validation messages (desired email is already in use, desired username is already in use, current password doesn't match what's in the DB)

Task #4 - Points: 1

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

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Updating Username/Email
<input type="checkbox"/> #2	1	Updating password
<input type="checkbox"/> #3	1	Don't just show code, translate things to plain English

Response:

Missing Response

Task #5 - Points: 1

Text: Include pull request links related to this feature

• Details:

Should end in /pull/#

URL #1

<https://github.com/rRahmanM6/rr42-it202-008/pull/30>

• Misc (1 pt.)

▲ COLLAPSE ▲

Task #1 - Points: 1

▼ EXPAND ▼

Text: Screenshot of wakatime

Task #2 - Points: 1

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

Task #3 - Points: 1

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

Task #4 - Points: 1

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

End of Assignment