Of course. Let's document **Practical 7**. This practical is about implementing a secure user session management system, which you have successfully done with your login, logout, and profile pages.

Here is a detailed guide on what to write in your practical file for Practical 7.

Practical 7: Create a Login/Logout System with Session & Cookie Handling

Aim: To implement a secure user authentication system for the ByteBazaar website using PHP sessions for state management, including login, logout, and redirection based on login status.

1. Key Questions & Analysis

1.1. Is the session securely started and terminated?

Yes. The system uses PHP's built-in session handling mechanisms to securely manage user sessions.

- **Session Start:** A session is started securely only after a user's credentials have been successfully verified. In login.php, session_start() is called, and the user's ID and other essential details are stored in the $_SESSION superglobal array. This happens only after password_verify() confirms the password is correct.
- **Session Termination:** The session is terminated explicitly in logout.php. The script starts the session, completely destroys all session data with session_destroy(), and then redirects the user to the homepage. This ensures that the user is securely logged out.

Example of Session Start from login.php:

PHP

```php
// From login.php
if (password_verify($password, $user['password'])) {
    // Set session
    session_start();
    $_SESSION['user_id'] = $user['u_id'];
    $_SESSION['first_name'] = $user['first_name'];
    // ... other session variables

    $response['success'] = true;
    $response['message'] = 'Login successful!';
}
```

1.2. Are cookies correctly managed?

PHP's session management handles the session cookie (PHPSESSID) automatically and securely. This cookie is sent to the user's browser upon login and is used to identify their session on subsequent requests. The cookie's lifetime is managed by the server configuration, ensuring it expires appropriately.

(Note: The "Remember me" checkbox from the supplementary problem is present in the HTML (login.html), but its backend logic for creating a long-lived cookie is a more advanced feature and has not been implemented in this phase.)

1.3. Is redirection based on login status implemented?

Yes. The application redirects or dynamically alters the UI based on the user's login status.

- **Login Success:** After a successful login, the frontend (js/login.js) receives a success message and redirects the user to the homepage (index.html).
- **Logout:** The logout.php script explicitly redirects the user to index.html using a header('Location: index.html') call after destroying the session.
- **Protected Pages:** Pages like profile.html rely on check_login.php. The JavaScript in js/profile.js calls this endpoint, and if the response indicates the user is not logged in, it redirects them to login.html.
- **Dynamic Navigation:** The js/script.js file also calls check_login.php to dynamically update the "Account" link in the navigation bar, changing it to a "Profile" link for logged-in users.

2. Test Cases (Post-Laboratory Work)

To demonstrate a working session system, include the following screenshots:

Test Case 1: Before Login
[Insert a screenshot of the homepage, showing the "Account" dropdown with "Login" and

"Register" options.]

Test Case 2: Successful Login

[Insert a screenshot of the login.html page after a successful login, showing the "Login successful!" notification.]

Test Case 3: After Login

[Insert a screenshot of the homepage after being redirected, now showing the "Account" link has changed to "Profile".]

Test Case 4: Session Inspection

[Open your browser's Developer Tools (F12), go to the "Application" (or "Storage") tab, and find the Cookies for your localhost. Take a screenshot that clearly shows the PHPSESSID cookie is present.]

Test Case 5: After Logout

[Insert a screenshot of the homepage after clicking "Logout," showing the "Account" link has reverted back to a dropdown with "Login" and "Register".]

This documentation thoroughly covers the implementation and verification of your session management system, fulfilling the requirements for Practical 7.