****

**ICT DEPARTMENT**

**DEVELOPMENT OPERATIONS(ITLDO801)**

**INSTRUCTIONS**

1. Make four groups

2. Each group will have one hour of presentation

3. copy all the commands together with their corresponding running screens into a Word document.

4. Submit the document through the email: bucyanny@gmail.com not later than Thursday, 30th Jan, 2025 at 8h30 AM. (one document for each group).

**PRACTICAL ASSIGNMENT NO3/30 marks**

1. **SCENARIO**

HEALTHRWANDA Ltd is a software company based in Kigali, Rwanda, with a focus on developing innovative solutions. Within its software engineering department, a proposal has been put forward to create a web application aimed at improving Rwanda's hospital management system. This application will enable patients view all their details such as diseases info, payment info, their doctor’s info to be contacted if necessary. However, to participate in a shared ride, patients must first register and log in to the system. As a DevOps Engineer at HEALTHRWANDA Ltd, you have been assigned the task of delivering the registration and login module for this proposed application.

1. **STEPS TO DELIVER THE MODULE**

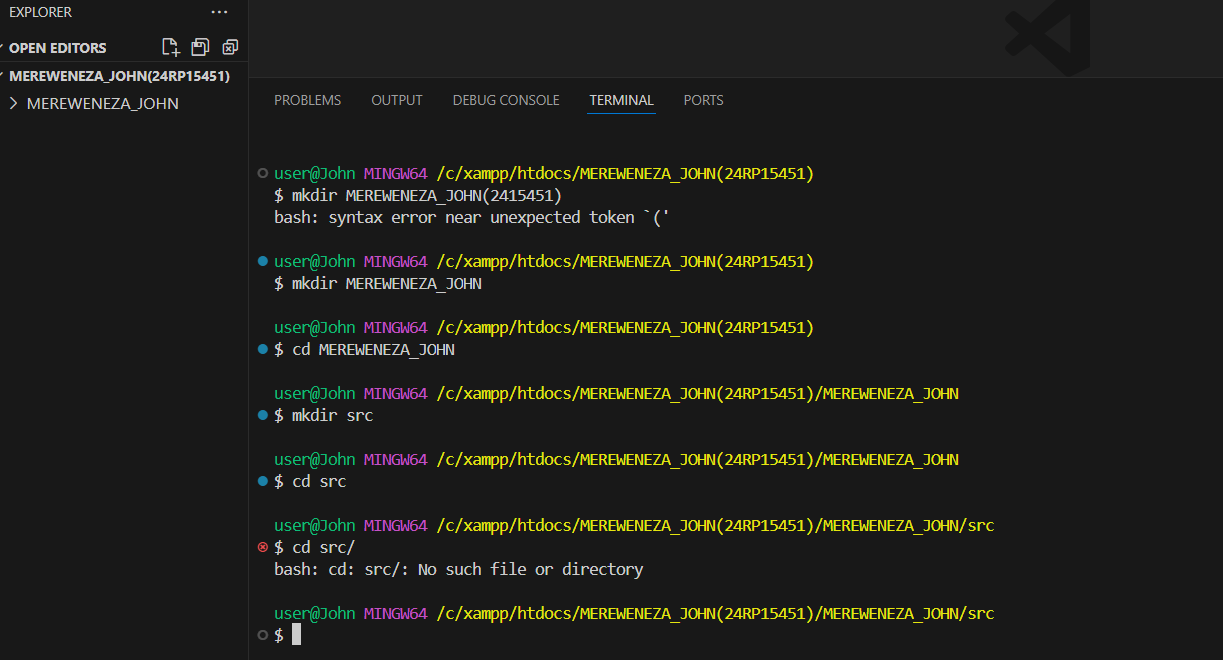
**Step 1: Create Project Folder**

Create a project folder named *YOUR\_GROUP\_NUMBER*.



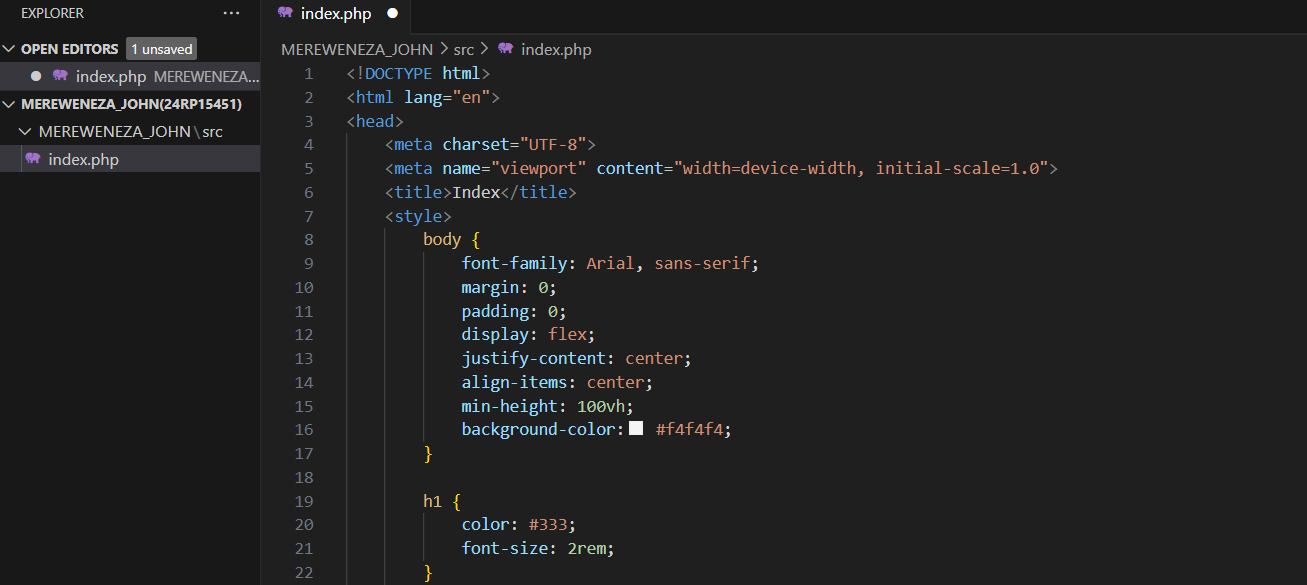
**Step 2: Create Src Folder**

Within the project folder, create a folder named *src* to store index, registration, and login PHP files.

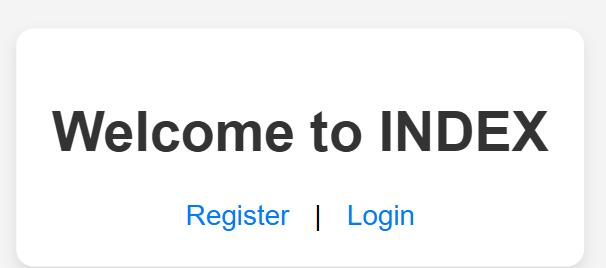


**Step 3: Create Index File**

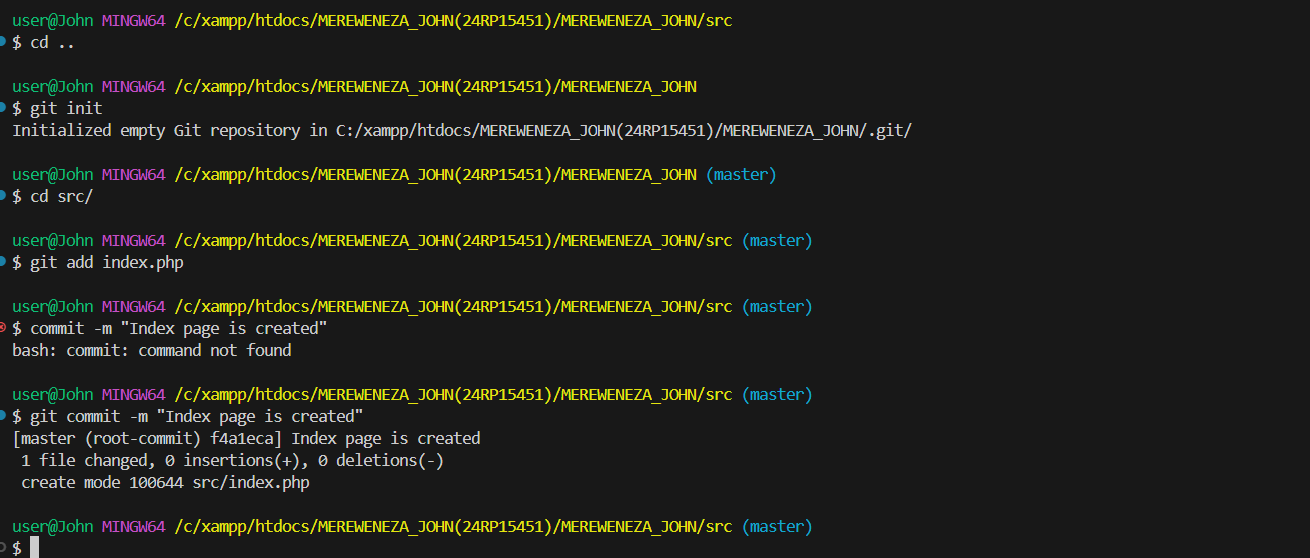
1. Inside the *src* folder, create an index file containing links to registration and login pages.



**OUTPUT:**

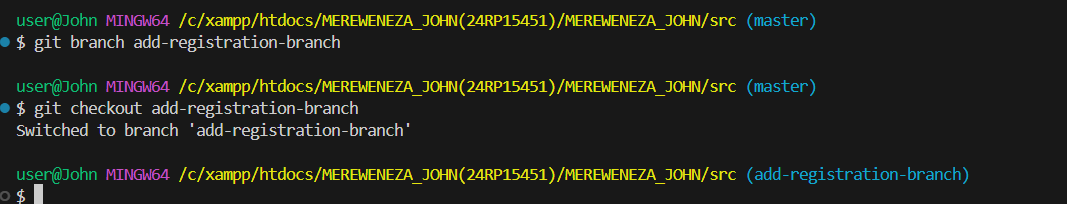


1. Commit changes with the message “Index page is created.”

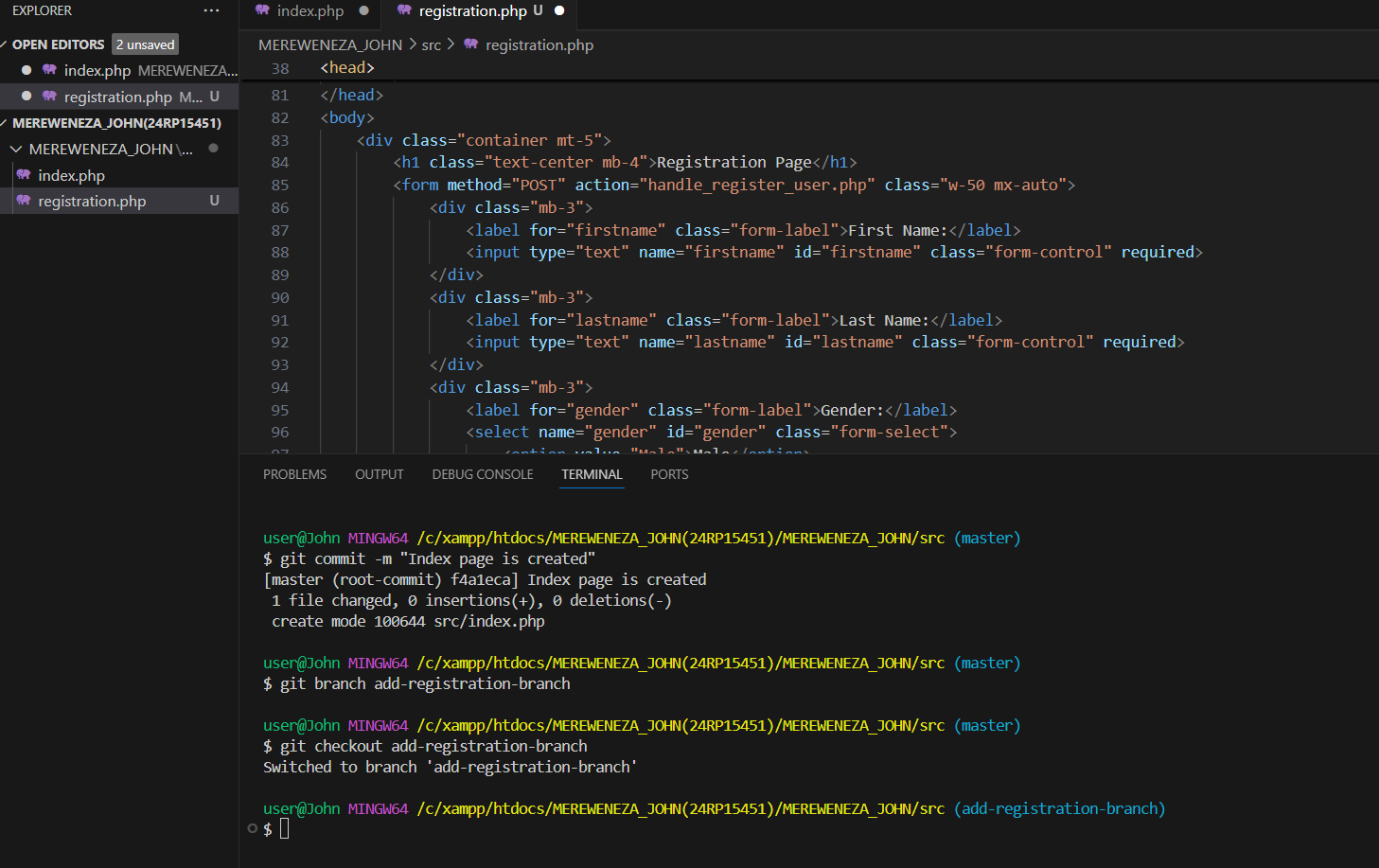


**Step 4: Create Registration Page**

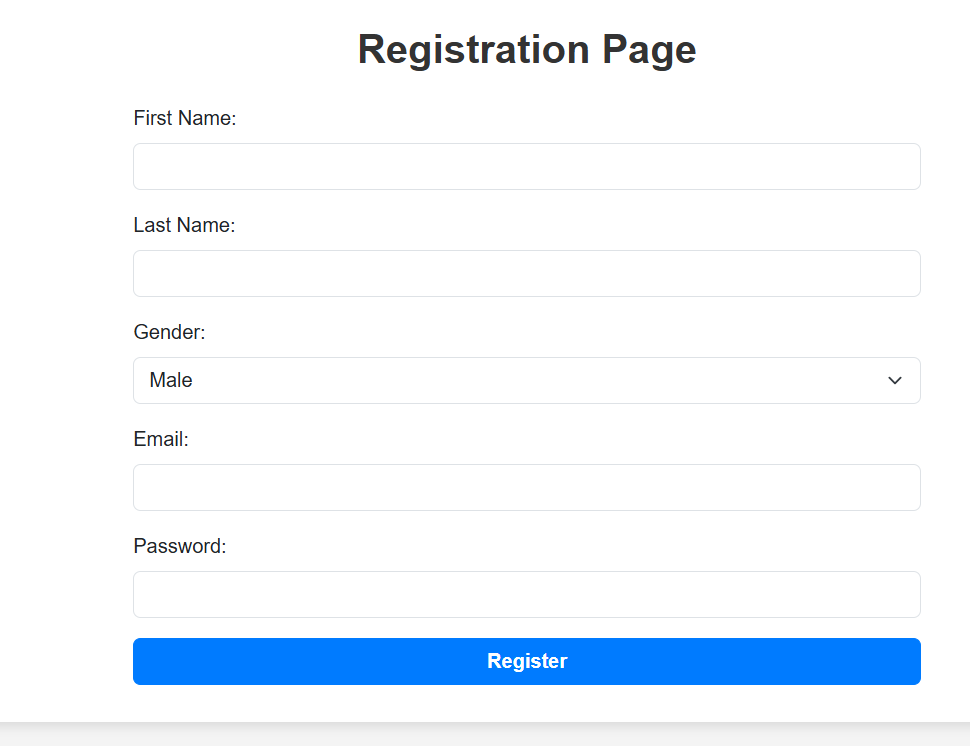
1. Create a new branch named *add-registration-branch.*



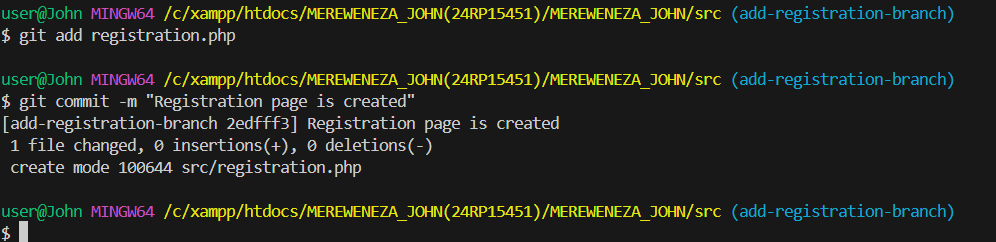
1. Add a registration page to the application under the *src* folder. Form fields are First name, Last name, Gender, Email and Password



**OUTPUT:**

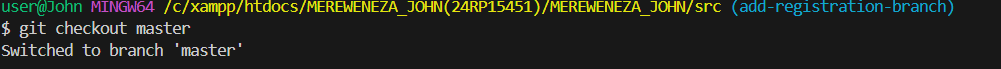


1. Commit changes with the message “Registration page is created.”

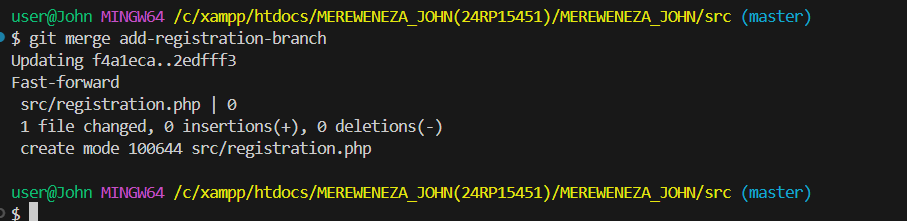


**Step 5: Merge Changes**

1. Switch to the main/master branch.

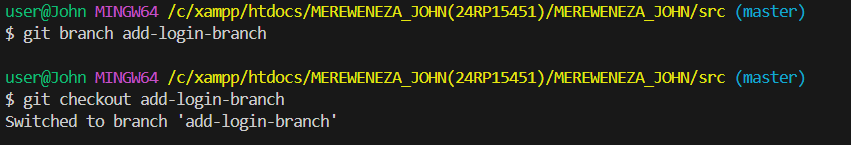


1. Merge changes made from the *add-registration-branch.*

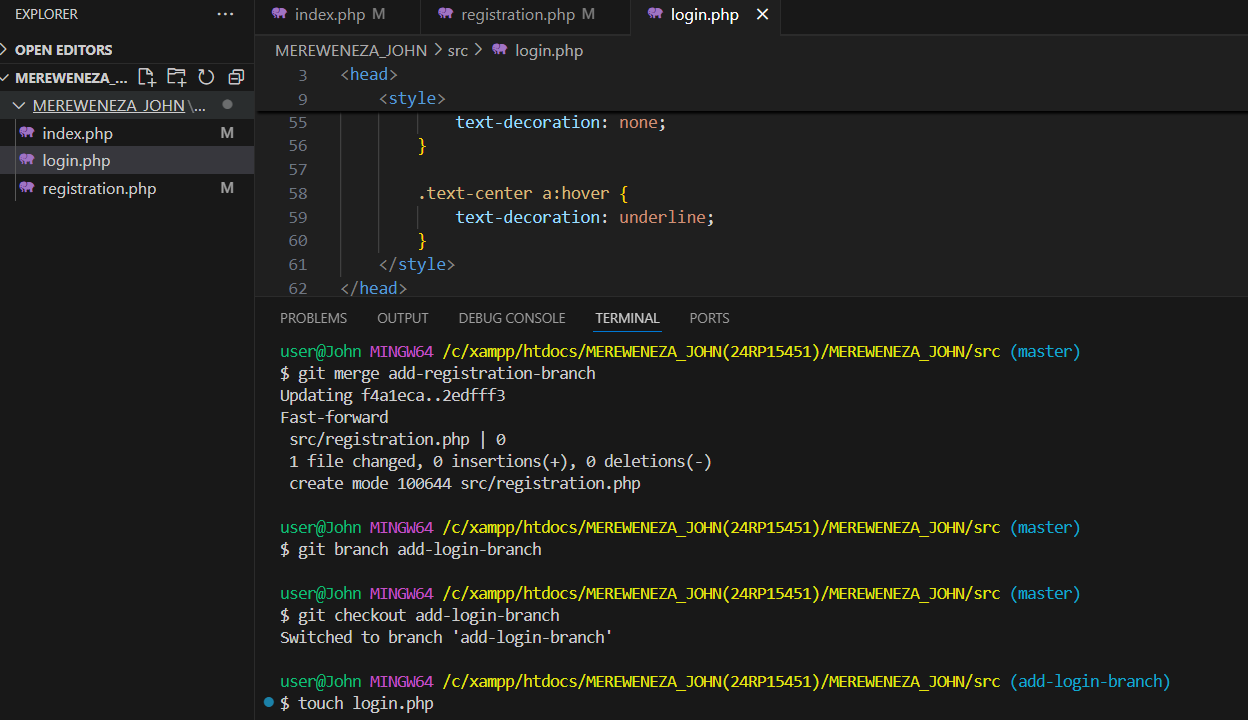


**Step 6: Create Login Page**

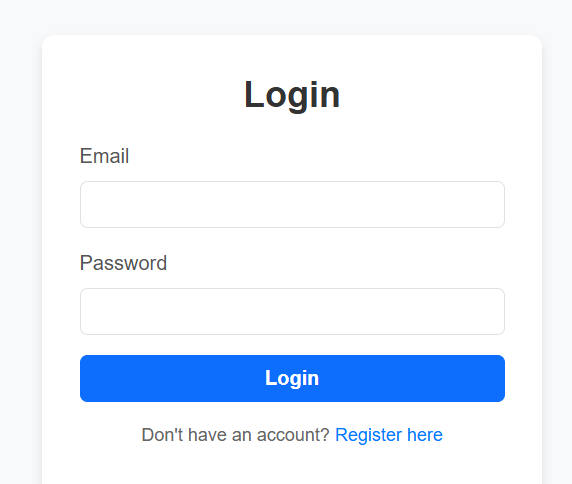
1. Create a new branch named *add-login-branch.*



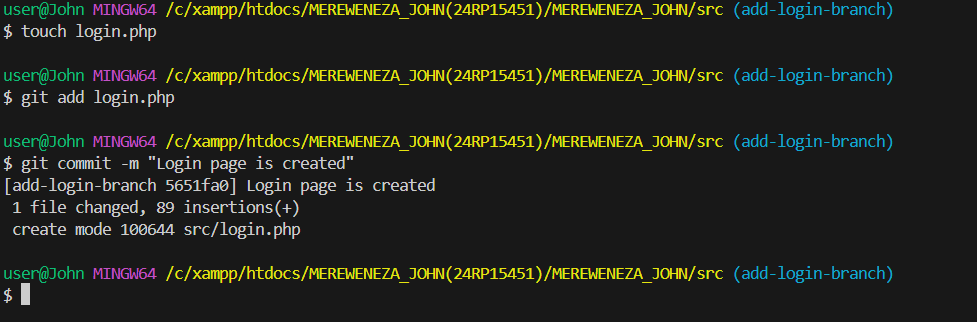
1. Add a login page to the application under the *src* folder. Login form fields are email and password



**OUTPUT**

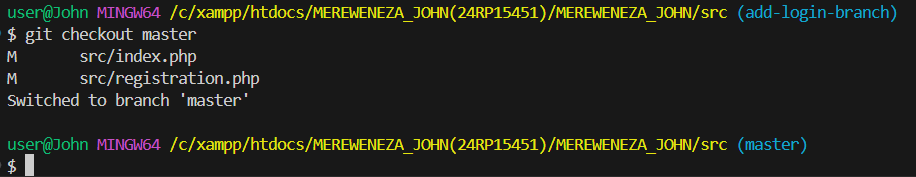


1. Commit changes with the message “Login page is created.”

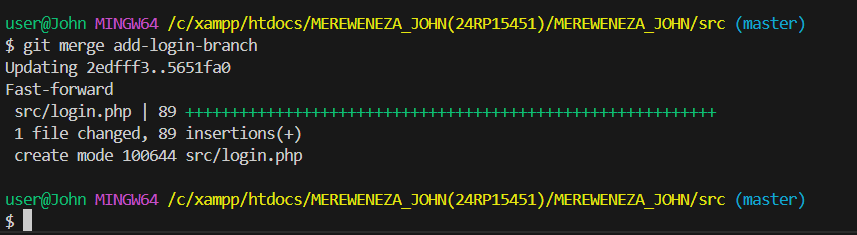


**Step 7: Merge Changes**

1. Switch to the main/master branch.

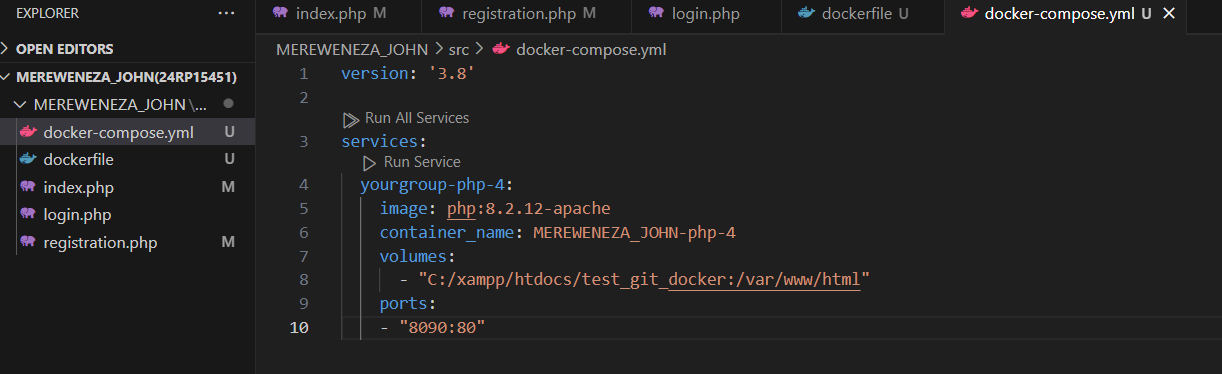


1. Merge changes made from the *add-login-branch.*

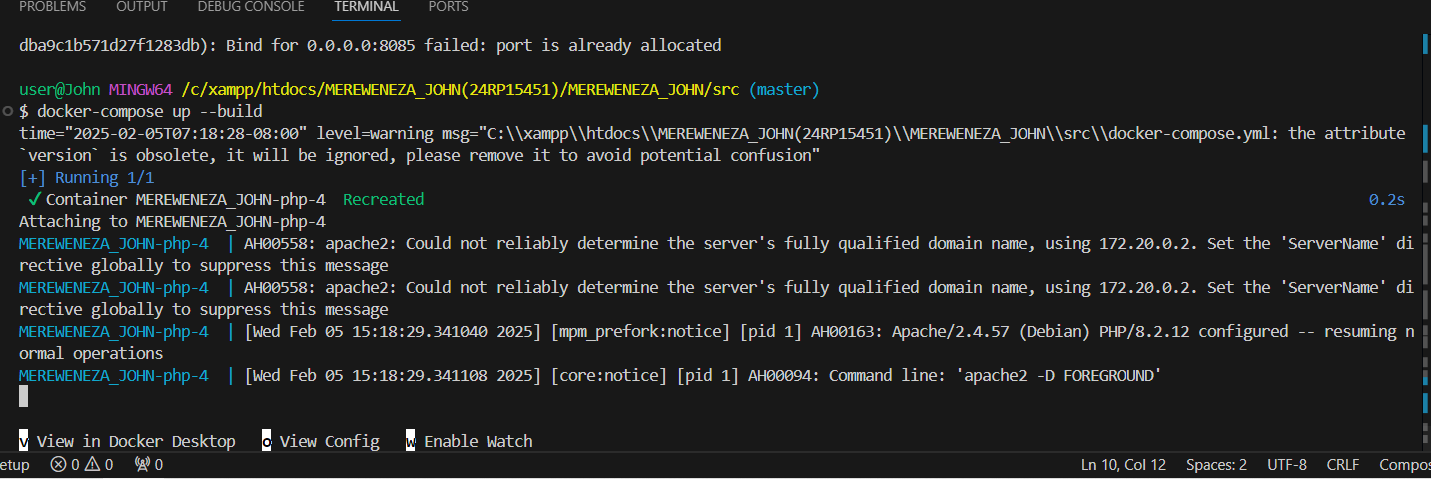


**Step 8: Dockerize the Module**

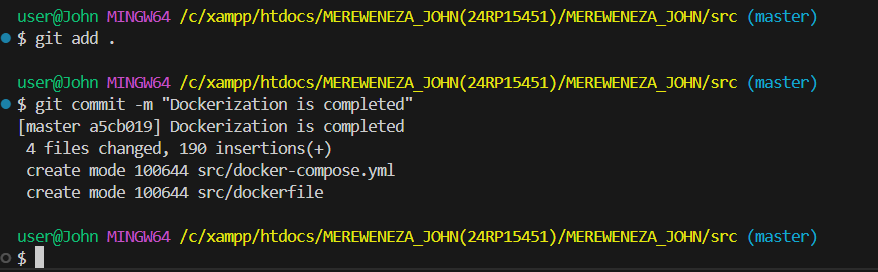
1. Define three services, with each service name beginning with *yourrgroupnumber*.



1. Dockerize the created module.

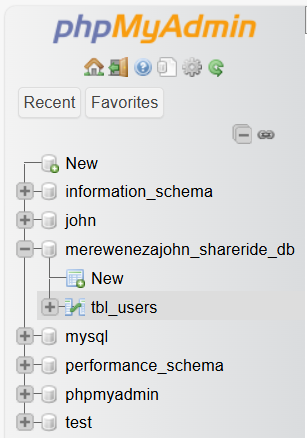


1. Commit all changes made in the module with the message “Dockerization is completed.”

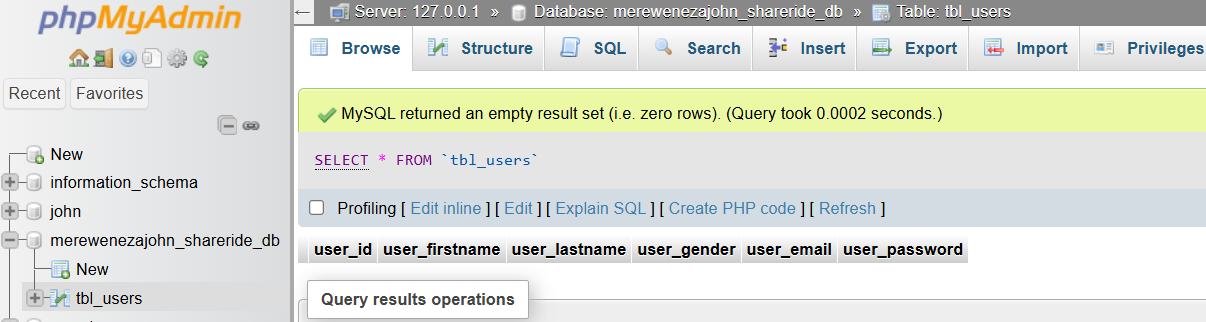


**Step 9: Create Database**

1. Create a database named *yourgroupnumber\_shareride\_db*.

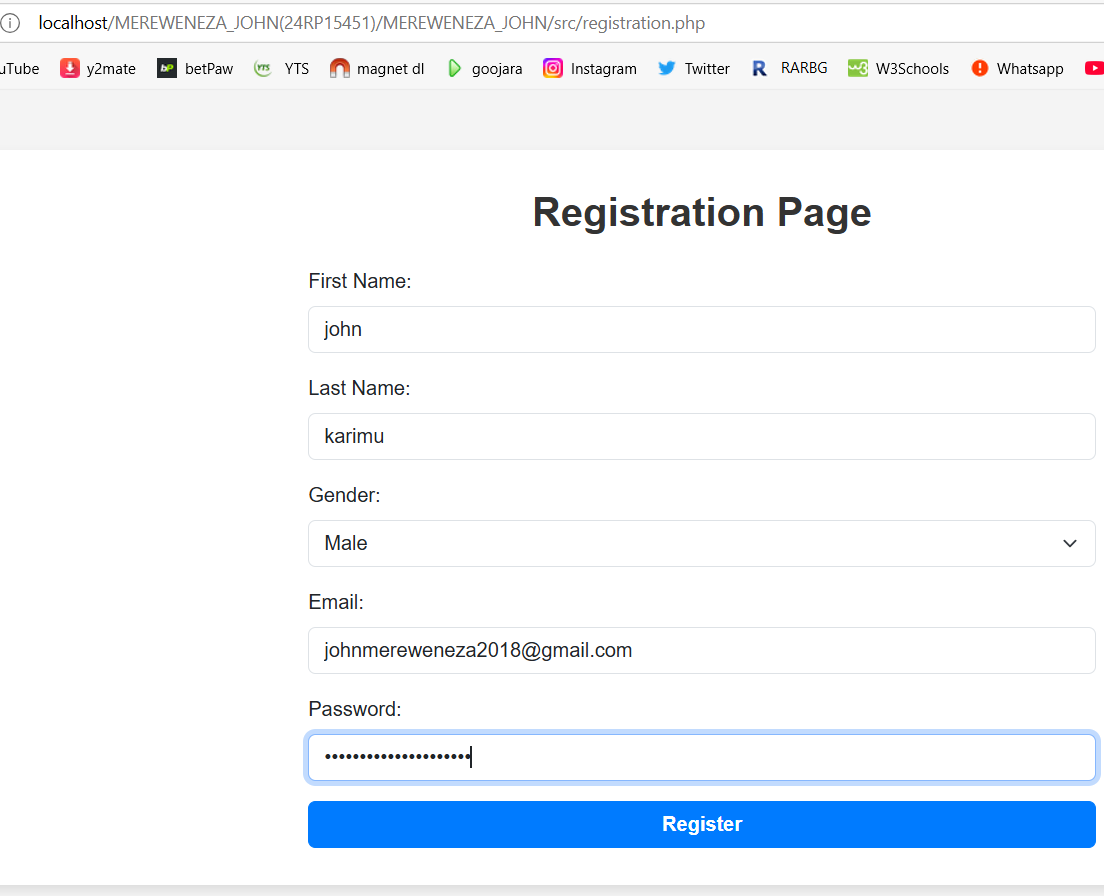


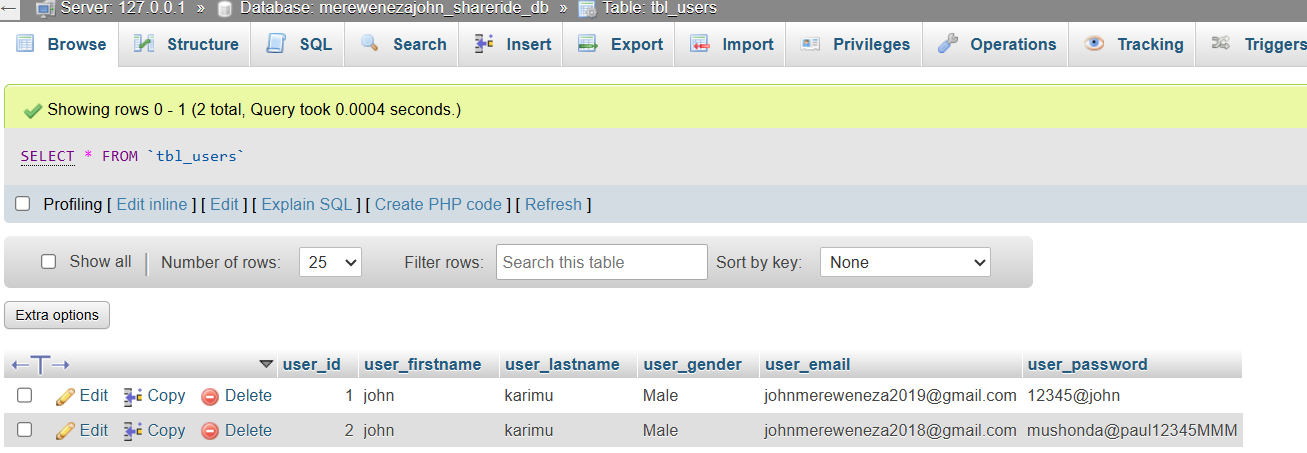
1. Add the table *tbl\_users* with the following fields: *user\_id, user\_firstname, user\_lastname, user\_gender, user\_email, user\_password*.



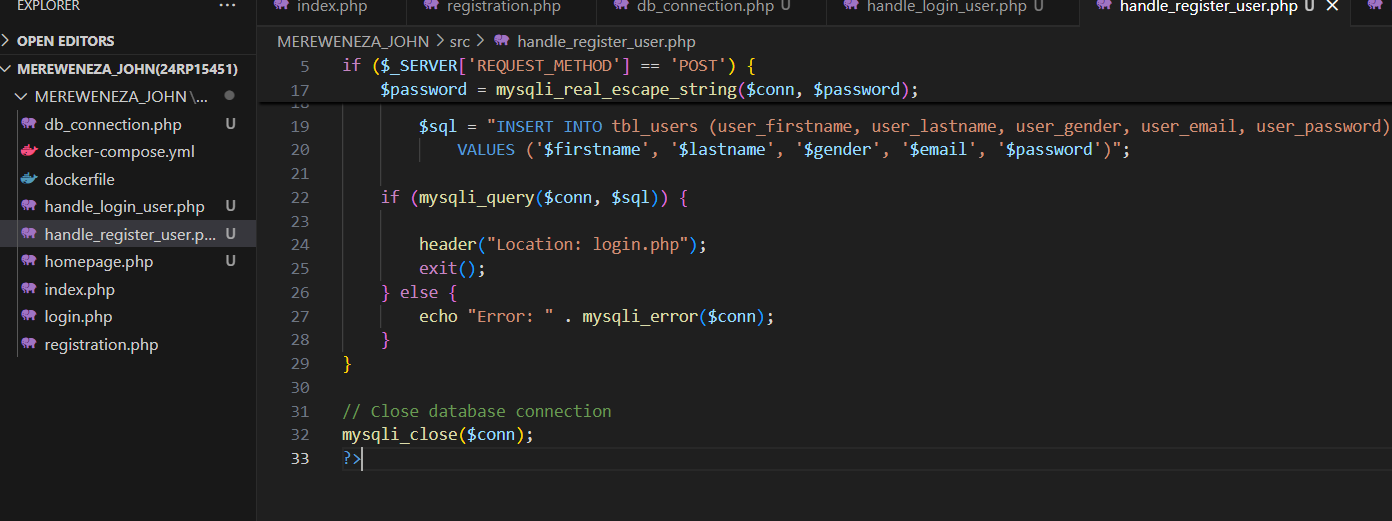
**Step 10: Registration Functionality**

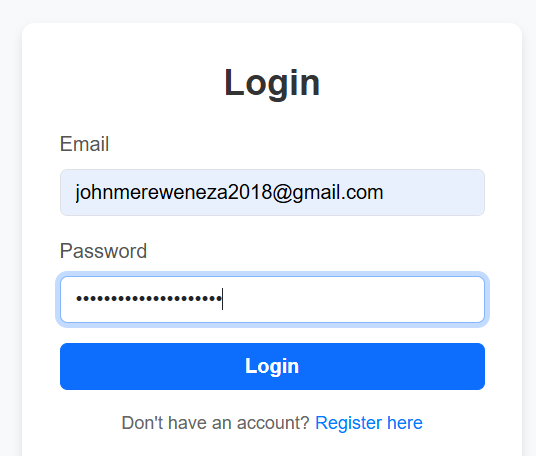
1. Add functionality to allow the user to complete registration.





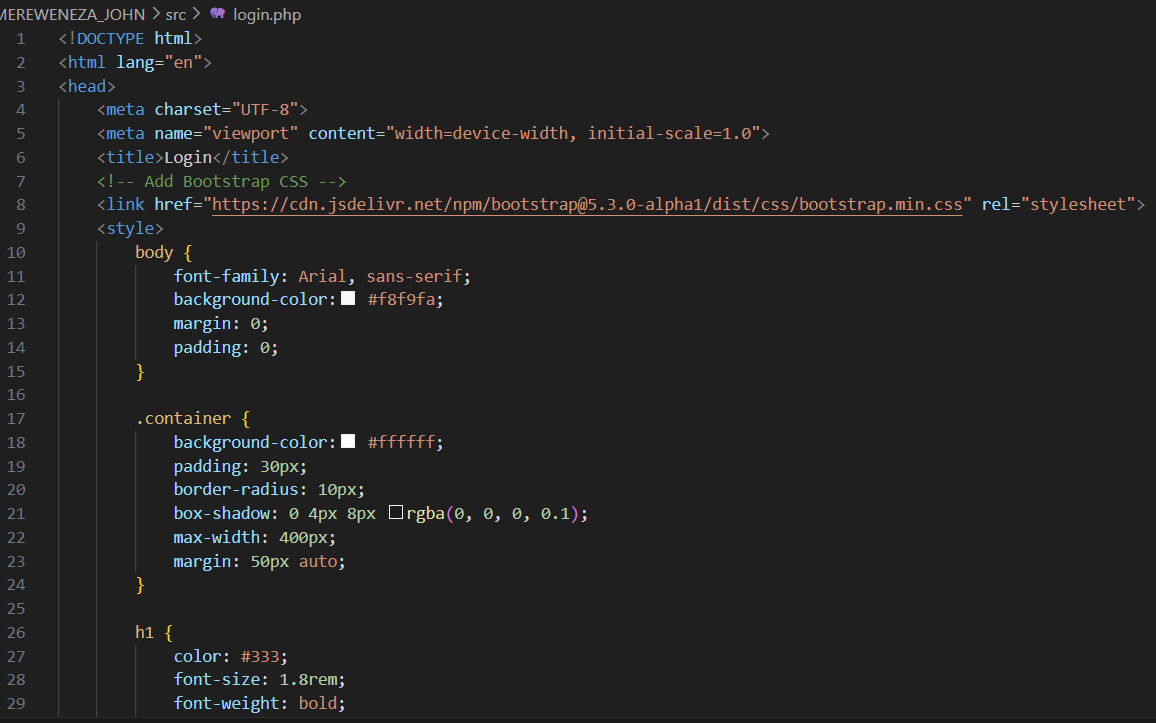
1. Once the user is registered, he/she will be redirected to login page



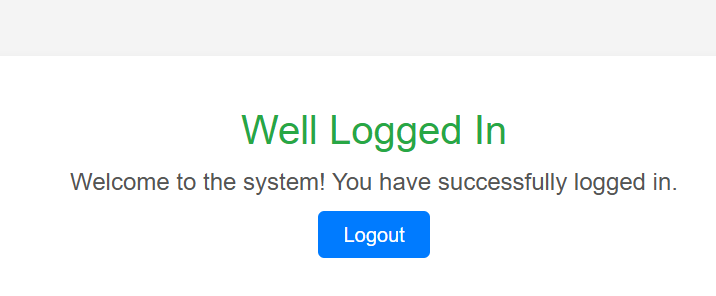


**Step 11: Login Functionality**

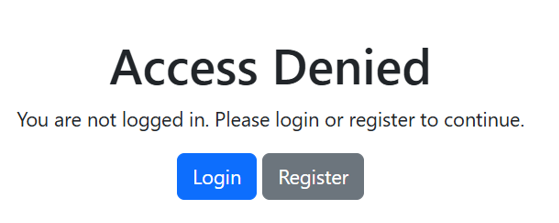
1. Add functionality to allow users to log in.



1. Once the user is well logged in, he/she will be redirected to home page that contains the message “Well logged in”

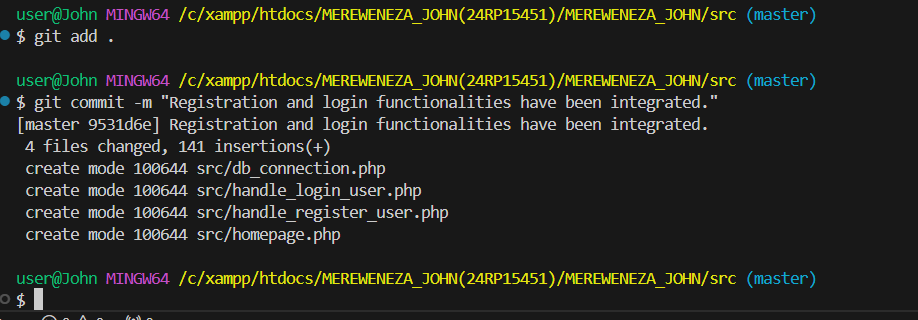


1. If the user is not logged in, he/she will be redirected to the page from where he/she will be able to open login page or registration page.



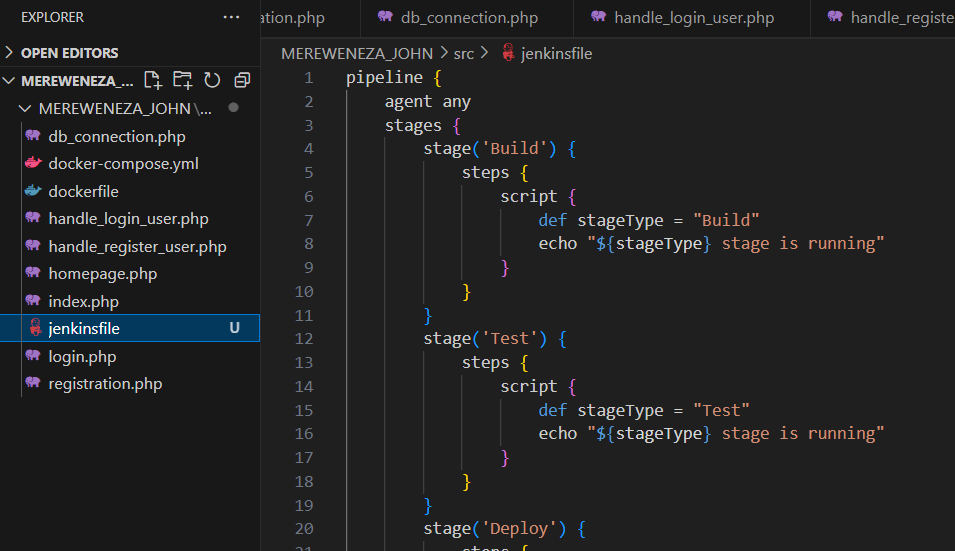
**Step 12: Commit Changes**

Commit changes made from steps 10 and 11 with the message “Registration and login functionalities have been integrated.”



**Step 13: Create Jenkins File**

1. Create a Jenkins file in the project folder.



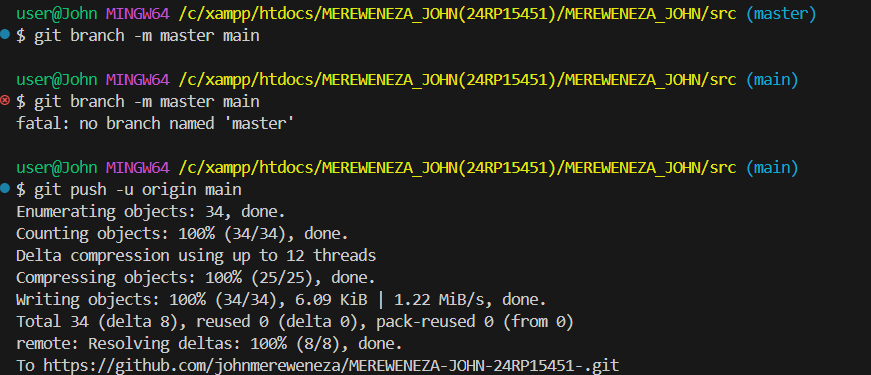
1. Attach source code showing different stages of DevOps.



1. Ensure every stage displays the message “**stage\_type** stage is running,” where the **stage\_type** represents the DevOps stage. Choose like 2 stages

**Step 14: Push module codes to Remote Repository**

Push the source code to the remote repository created on GitHub.



**Step 15: Jenkins Pipeline Creation and testing**

Using Jenkins, create and test a pipeline using SCM of type Git.

Push

Git add.

Git commit –m “your comment”

git remote add origin <https://github.com/johnmereweneza/MEREWENEZA-JOHN-24RP15451-.git>

git branch -m master main

git push -u origin main