SWINBURNE UNIVERSITY OF TECHNOLOGY

SYDNEY CAMPUS

MASTER OF INFORMATION TECHNOLOGY (PROFESSIONAL COMPUTING)

COS80021-Web Application Development

Task: Lab01

Submitted to

Dr. Mohammad Ahmadi

Submitted by

Santosh Pokhrel

Student ID: 104053011

Date: 02-02-2024

February 2024

1. Copy the code from “get data with PHP” example from Canvas into your mercury account. Load them into your preferred editor, and just read the files carefully, so that you get a good feeling for the structure of the example, and appreciate how the html and PHP technologies work together. In both IE and Firefox, go to the URL of the PHP file (simplephp.php) and run the PHP example. The PHP program has a “sleep” statement in it, so that the server takes time to respond.

Answer: As per the instructions, I copied the given codes from the “get data with PHP” folder in VSCode and I read the code carefully. I found that there is one php file called simplephp.php which contains one html form with username and password fields. And there are few lines of php code to get the user input data from the form fields with the help of $_GET function as :

```
$name = $_GET['namefield'];

$pwd = $_GET['pwdfield'];
```

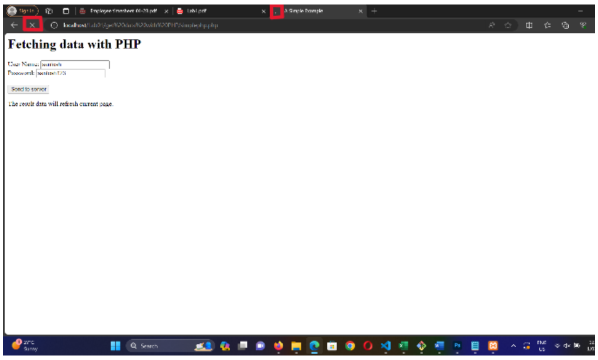
and the sleep time is set to 10 seconds to delay the data fetching process as:

```
sleep(10);
```

and in the end, the the username and passwords are displayed with the help of echo function as:

```
ECHO ($name." : ".$pwd);
```

I noticed that while the data is fetching from the server I was not able to interact with the web page, the refresh icon was loading for 10 second and could not perform any other tasks.



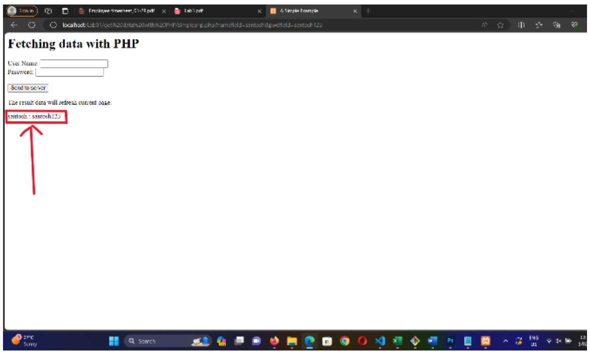
Fetching data with PHP

User Name: santosh
Password: santosh123

Send to server

The result data will refresh current page.

Sign in
Employee timesheet 01-28.pdf
lab1.pdf
A Simple Example
localhost/Lab01/get%20data%20with%20PHP/simplephp.php



Fetching data with PHP

User Name:
Password:

Send to server

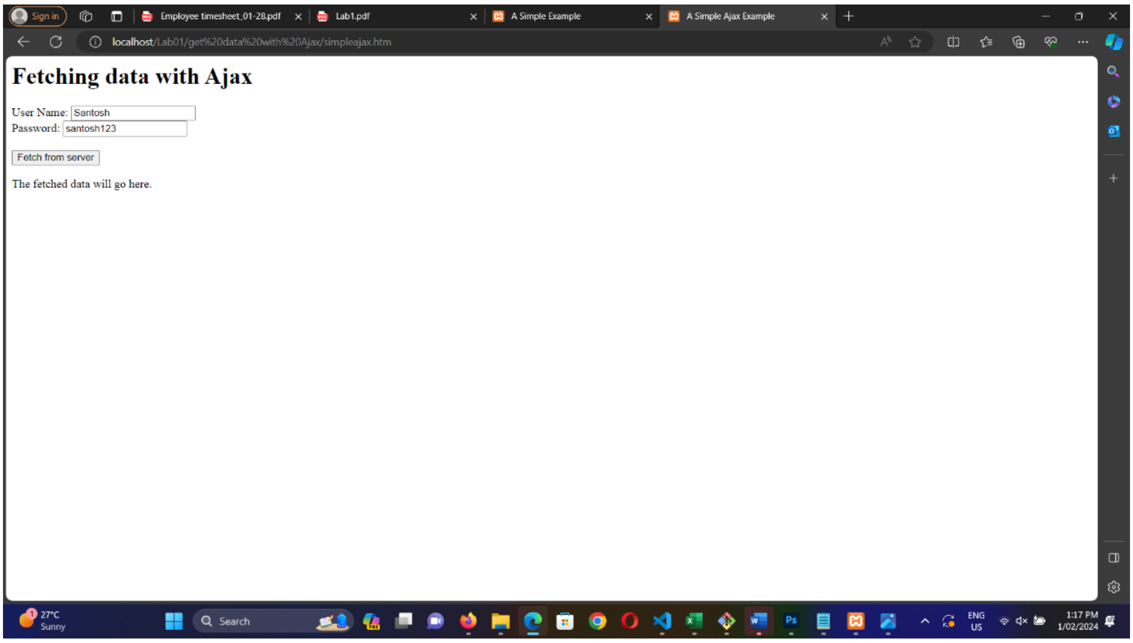
The result data will refresh current page.

santosh : santosh123

Sign in
Employee timesheet_01-28.pdf
Lab1.pdf
A Simple Example
localhost/Lab01/get%20data%20with%20PHP/simplephp.php?namefield=santosh&pwdfield=santosh123

2. Copy the code of “get data with Ajax” example from Canvas, into your mercury account. Load them into your preferred editor, and just read the files carefully, so that you get a general picture for the structure of the example, and appreciate how the html, JavaScript and PHP technologies work together with Ajax. In both IE and Firefox, go to the URL of the HTML files and run the Ajax example (simpeajax.htm). The PHP program has a “sleep” statement in it, so that the server takes time to respond. Experiment with various client interactions in the waiting period – for example, try entering different data whilst waiting for the server to respond, and see what the eventual response is.

Answer: I copied the code and run simpleajax.htm file, upon running I found that the data was fetched without refreshing the page and another important this is that I was able to input the new data after submitting the current data which was not possible before while fetching the data without the use of ajax.



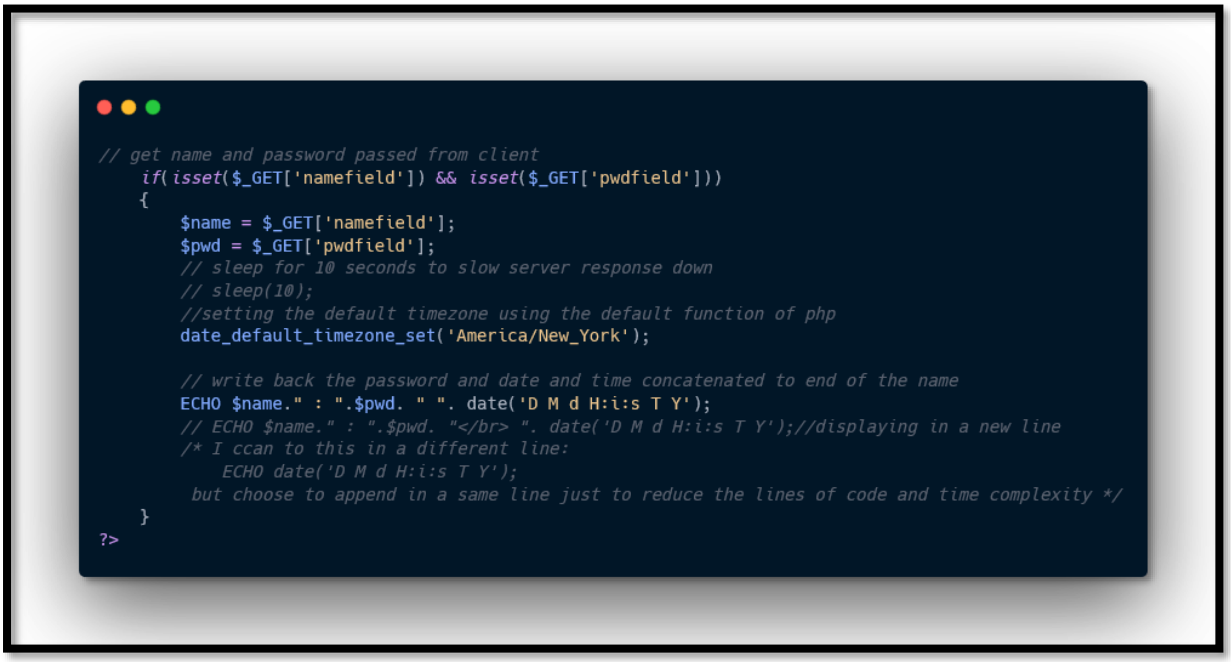
Fetching data with Ajax

User Name: Santosh
Password: santosh123

The fetched data will go here.

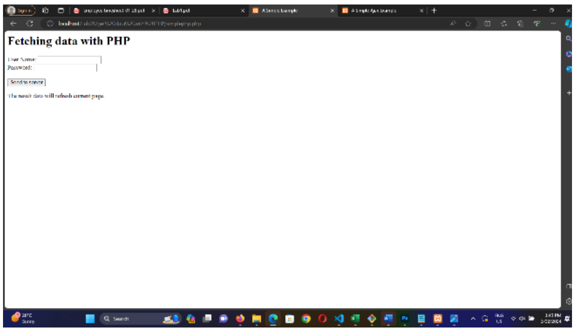
3. We now add a new feature based on 1. Modify the “simplephp.php” page so that it also displays current server time on the page after the user click ‘Send to server’ button. Use PHP built-in functions to finish this task.

Answer: I used date_default_timezone_set to set the default timezone and then echo date('D M d H:i:s T Y'); to display the current date and time.



```
// get name and password passed from client
    if(isset($_GET['namefield']) && isset($_GET['pwdfield']))
    {
        $name = $_GET['namefield'];
        $pwd = $_GET['pwdfield'];
        // sleep for 10 seconds to slow server response down
        // sleep(10);
        //setting the default timezone using the default function of php
        date_default_timezone_set('America/New_York');

        // write back the password and date and time concatenated to end of the name
        ECHO $name." : ".$pwd. " ". date('D M d H:i:s T Y');
        // ECHO $name." : ".$pwd. "</br> ". date('D M d H:i:s T Y');//displaying in a new line
        /* I ccan to this in a different line:
            ECHO date('D M d H:i:s T Y');
         but choose to append in a same line just to reduce the lines of code and time complexity */
    }
?>
```

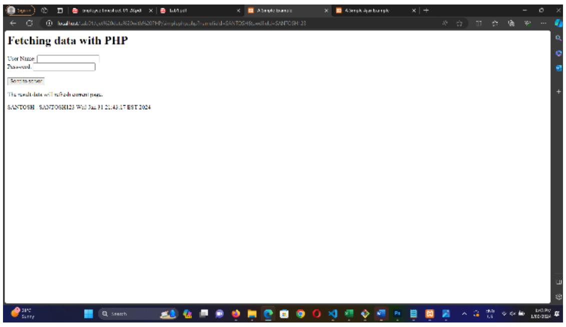
OUTPUT:



Fetching data with PHP

User Name:
Password:

The result data will refresh current page.

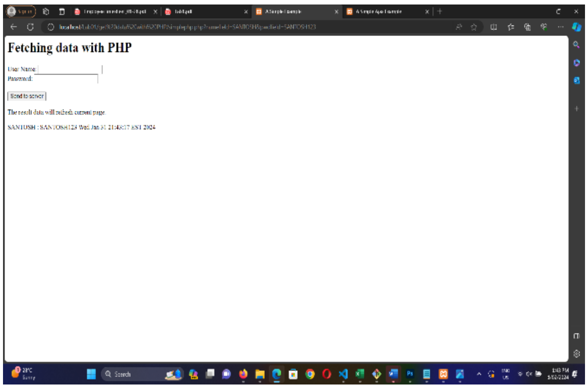


Fetching data with PHP

User Name:
Password:

The result data will refresh current page.

SANTOSH : SANTOSH123 Wed Jan 31 21:43:17 EST 2024



Fetching data with PHP

User Name:
Password:

The result data will refresh current page.

SANTOSH : SANTOSH123 Wed Jan 31 21:43:17 EST 2024

4. Modify the “simpeajax.js” file in c). Change the third parameter of ‘XMLHttpRequest.open’ function from “true” to “false”, i.e., change ‘xhr.open("GET", url, true)’ to ‘xhr.open("GET", url, false)’. Then run both examples and try to find the difference between them. (Hint: The difference between these two examples is due to asynchronous/synchronous mode of Ajax).

Answer:

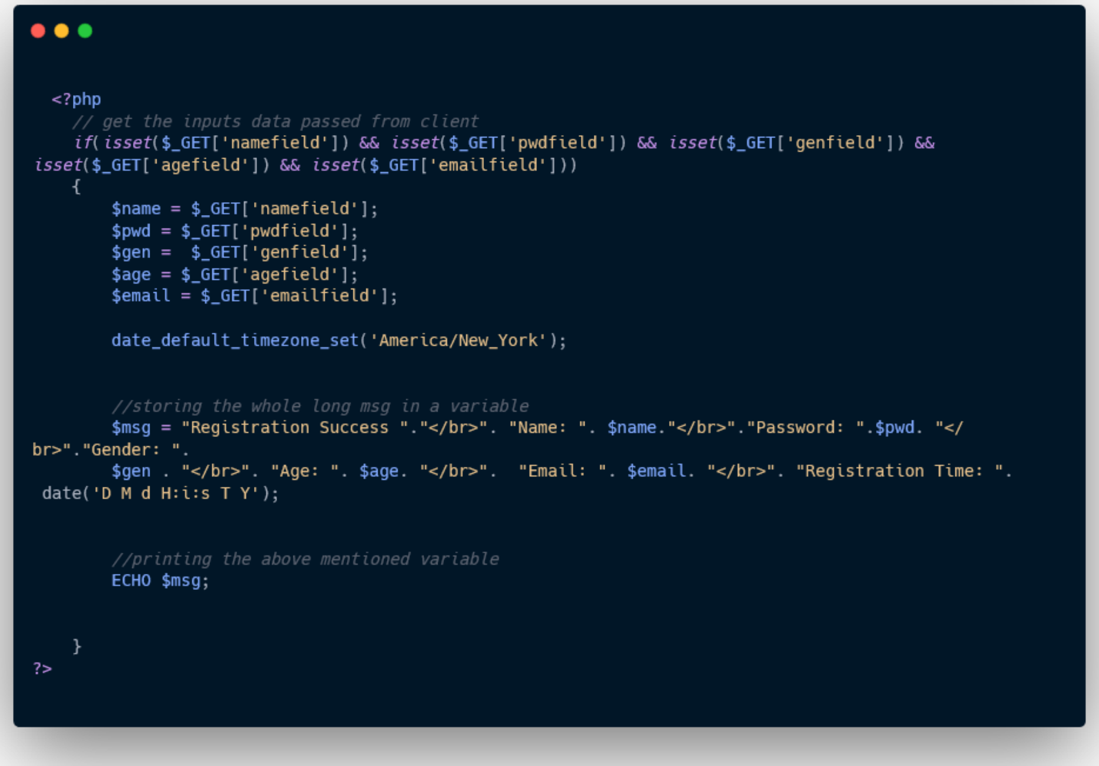
- **Asynchronous (original version with true)**: The browser doesn't wait for the AJAX call to complete and continues to process other tasks. The user interface remains responsive. When the request completes, the onreadystatechange function is triggered, and the response is processed.
- **Synchronous (modified version with false)**: The browser waits and does not process any other task until the AJAX call completes. This means the user interface can become unresponsive or "freeze" until the request finishes.

Additional exercise

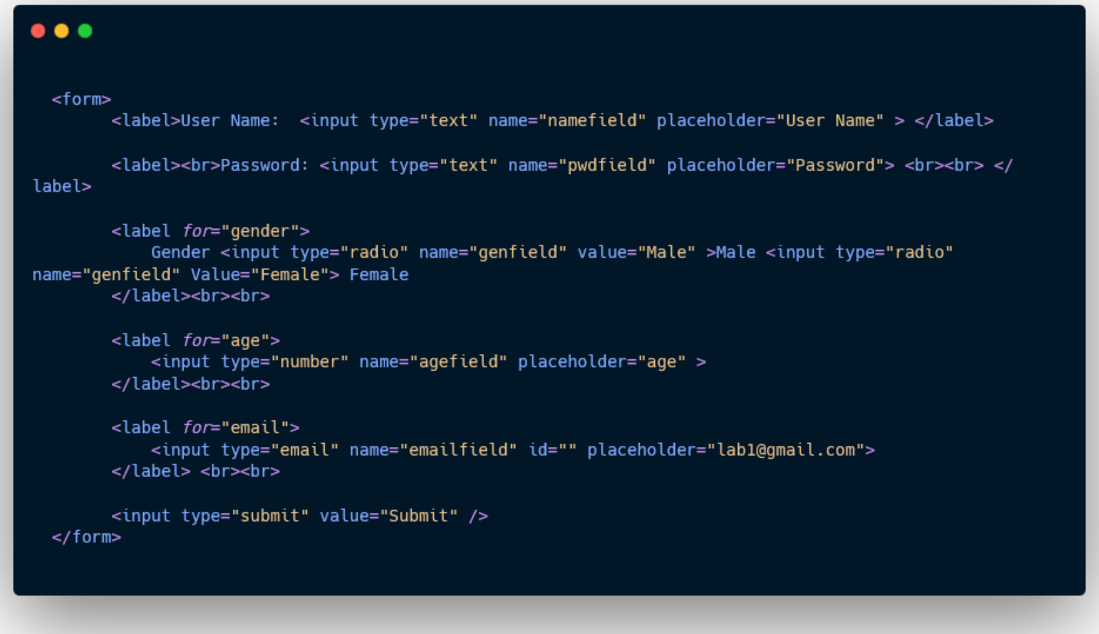
5. Modify the “simplephp.php” file. We now try to extend this page to a user registration page. 1) Change the file name to ‘userRegistration.php’. 2) Use HTML form inputs to get user information (see Figure 2). We ask users to input their Username/Password/Gender/Age Range/Email. Try to use different form input types such as text/password/ 3) When the user click ‘submit’ button, the PHP page simply check whether user input all required information. If all the information exists, the PHP page displays all input data with the registration time to users.

Answer:

Code:

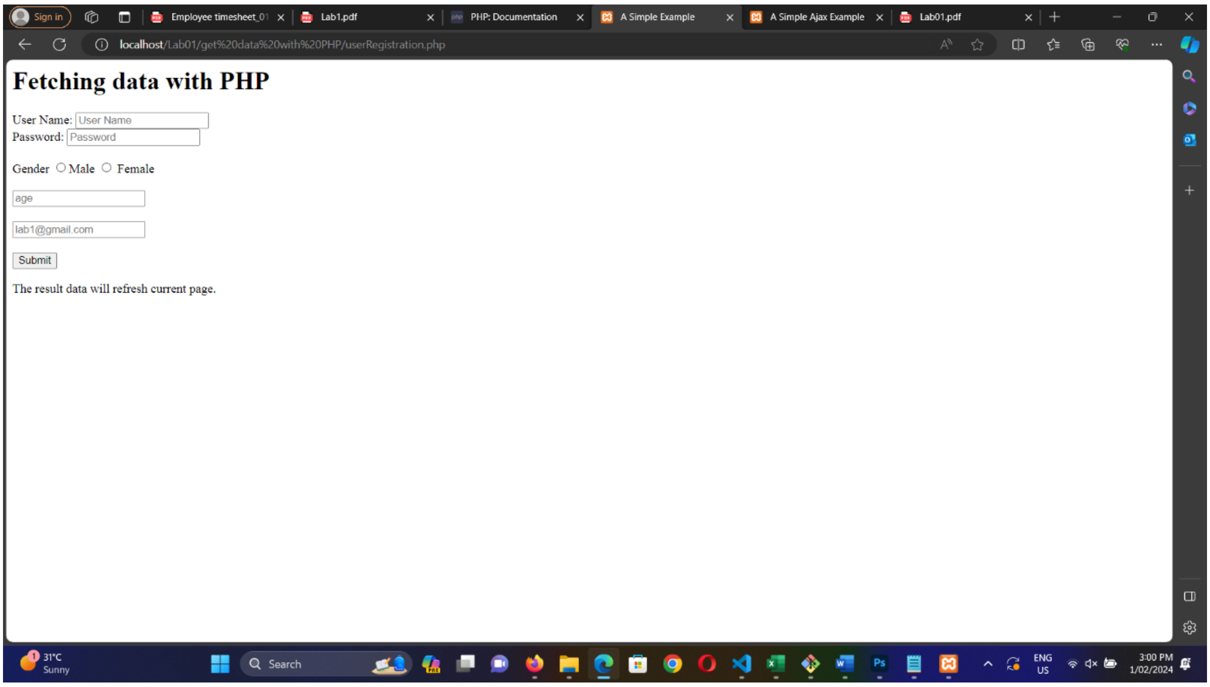


```
  <?php
    // get the inputs data passed from client
    if( isset($_GET['namefield']) && isset($_GET['pwdfield']) && isset($_GET['genfield']) &&
isset($_GET['agefield']) && isset($_GET['emailfield']))
    {
        $name = $_GET['namefield'];
        $pwd = $_GET['pwdfield'];
        $gen =  $_GET['genfield'];
        $age = $_GET['agefield'];
        $email = $_GET['emailfield'];

        date_default_timezone_set('America/New_York');


        //storing the whole long msg in a variable
        $msg = "Registration Success "."</br>". "Name: ". $name."</br>"."Password: ".$pwd. "</
br>"."Gender: ".
        $gen . "</br>". "Age: ". $age. "</br>".  "Email: ". $email. "</br>". "Registration Time: ".
 date('D M d H:i:s T Y');


        //printing the above mentioned variable
        ECHO $msg;


    }
?>
```



```
  <form>
        <label>User Name:  <input type="text" name="namefield" placeholder="User Name" > </label>

        <label><br>Password: <input type="text" name="pwdfield" placeholder="Password"> <br><br> </
label>

        <label for="gender">
            Gender <input type="radio" name="genfield" value="Male" >Male <input type="radio"
name="genfield" Value="Female"> Female
        </label><br><br>

        <label for="age">
            <input type="number" name="agefield" placeholder="age" >
        </label><br><br>

        <label for="email">
            <input type="email" name="emailfield" id="" placeholder="lab1@gmail.com">
        </label> <br><br>

        <input type="submit" value="Submit" />
  </form>
```

Output:



Fetching data with PHP

User Name: User Name
Password: Password

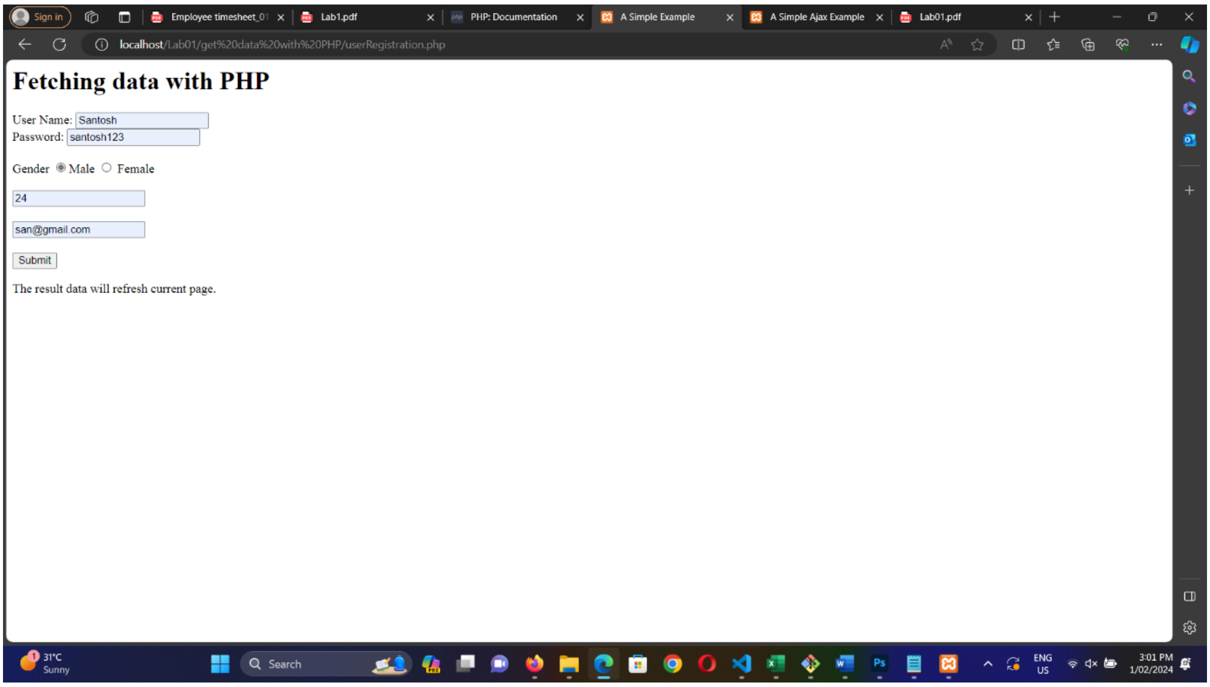
Gender ○ Male ○ Female

age

lab1@gmail.com

Submit

The result data will refresh current page.



Fetching data with PHP

User Name: Santosh
Password: santosh123

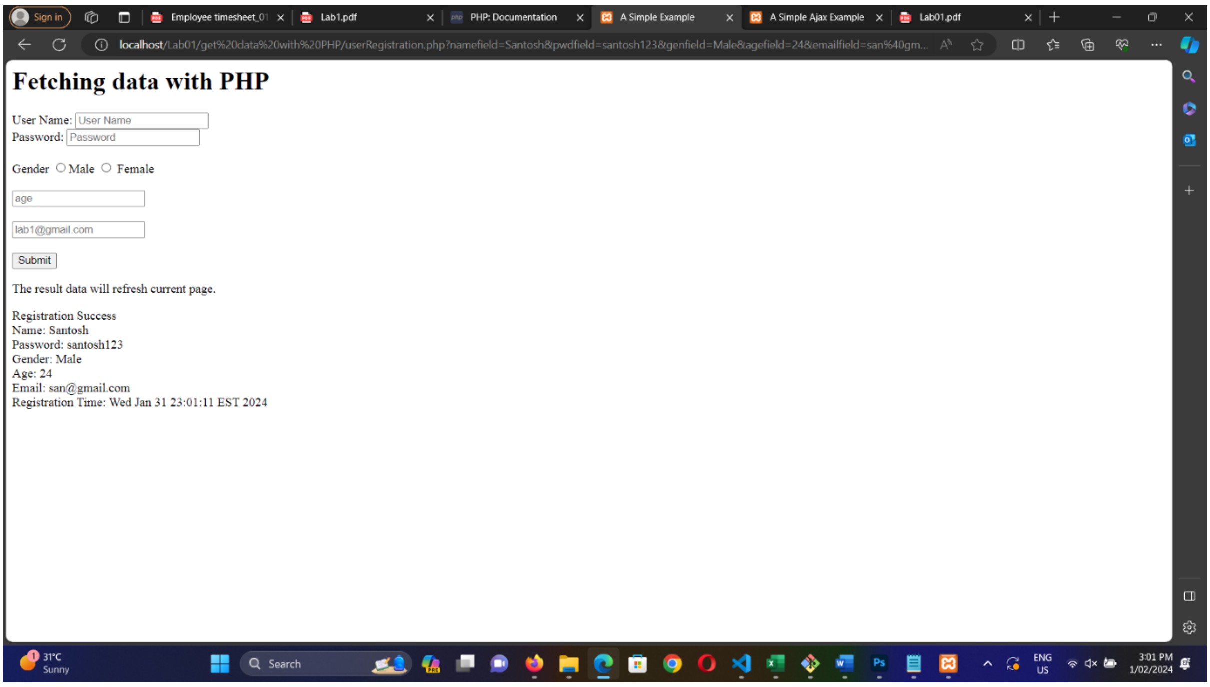
Gender ● Male ○ Female

24

san@gmail.com

Submit

The result data will refresh current page.



Sign in
Employee timesheet_01
Lab1.pdf
PHP: Documentation
A Simple Example
A Simple Ajax Example
Lab01.pdf
localhost/Lab01/get%20data%20with%20PHP/userRegistration.php?namefield=Santosh&pwdfield=santosh123&genfield=Male&agefield=24&emailfield=san%40gm...
31°C
Sunny
Search
ENG
US
3:01 PM
1/02/2024

Fetching data with PHP

User Name: User Name
Password: Password

Gender ○ Male ○ Female

age

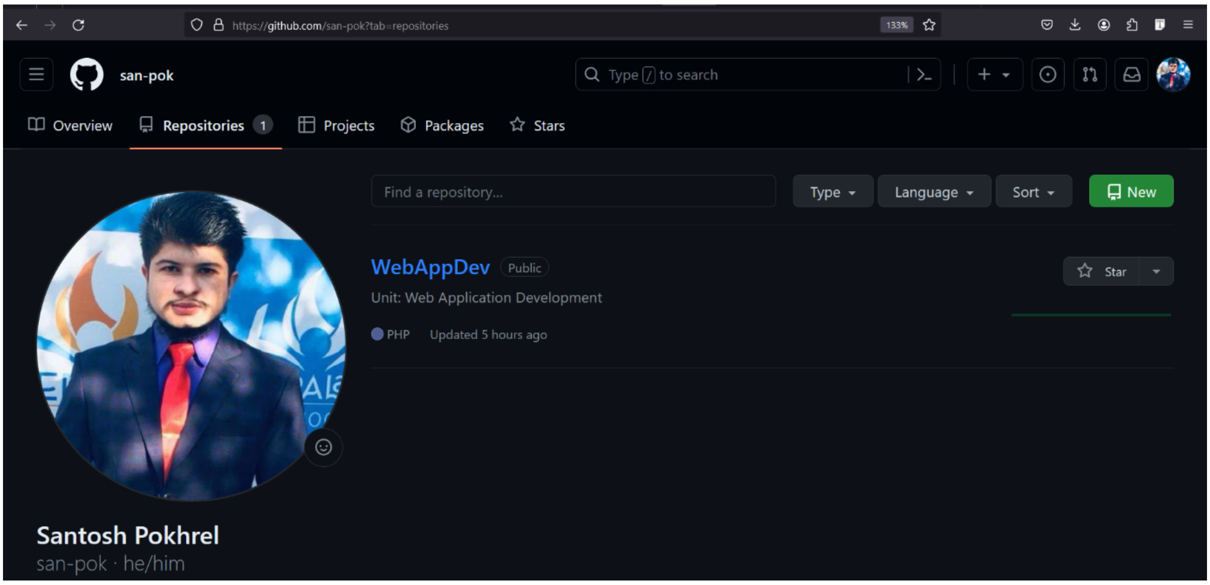
lab1@gmail.com

Submit

The result data will refresh current page.

Registration Success
Name: Santosh
Password: santosh123
Gender: Male
Age: 24
Email: san@gmail.com
Registration Time: Wed Jan 31 23:01:11 EST 2024

Note: Code is available on my github account in a repository called: WebAppDev



WebAppDev Public
Unit: Web Application Development
PHP Updated 5 hours ago

Santosh Pokhrel
san-pok · he/him

References

[1] PHP documentation. Available PHP: Documentation