

# CST 226-2 Web Application Development

## Lesson 04: Working with Databases

### Activity 09: Utilizing PDO for Database Operations

#### Instructions:

1. Design an HTML page as shown in Figure 1, allowing users to input information for user registration in a small web application.

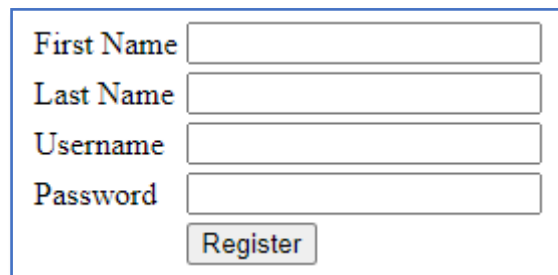


Figure 1 shows a registration form with four input fields and a button. The fields are labeled 'First Name', 'Last Name', 'Username', and 'Password'. The 'Register' button is located below the 'Password' field.

Figure 1: HTML page

2. After designing the HTML page, modify the HTML to send the user inputs to the "register.php" page.
3. Ensure that the password is not displayed as plain text. Implement appropriate techniques, such as using the password\_hash function, to make it unreadable to humans.
4. The output generated by your PHP page should resemble Figure 2.

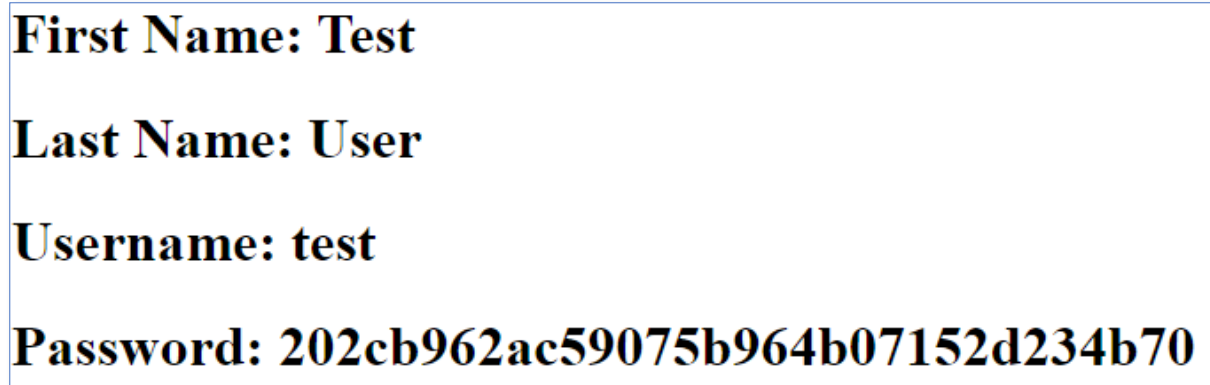


Figure 2 shows the output of the PHP page, displaying the registration details in a bold, black font. The output is as follows:

**First Name: Test**

**Last Name: User**

**Username: test**

**Password: 202cb962ac59075b964b07152d234b70**

Figure 2: PHP Page Output

5. Create the DbConnector class using PDO in an appropriate manner.
6. Create the necessary database structure as follows.

```
CREATE TABLE user (  
    id int(11) NOT NULL AUTO_INCREMENT,  
    firstname varchar(50) NOT NULL,  
    lastname varchar(50) NOT NULL,
```

```
username varchar(50) NOT NULL,  
password text NOT NULL,  
PRIMARY KEY (id)  
);
```

7. Save the user inputs to the database using the prepared statement approach.
8. If the information is successfully saved to the database, display a message to the user saying "Successfully Saved."
9. If any errors occur during this process, inform the user with an "Error Occurred" message.
10. Add a few data to the database using the above interface.
11. Now, implement a "view.php" page to view the users already registered with the system.
12. Display users one by one, showing their first name, last name, and some specific actions as shown in Figure 3.

No.	First Name	Last Name	Action
1	Test	User	<a href="#">Edit</a>   <a href="#">Delete</a>
2	Subhash	Ariyadasa	<a href="#">Edit</a>   <a href="#">Delete</a>
3	Senuja	Nimsith	<a href="#">Edit</a>   <a href="#">Delete</a>

Figure 3: view.php Page Output

13. Once user clicks on this Edit action, "edit.php" page should load the specific user information in edit mode.
14. Once the edit is success, the updated information should be saved to the database.
15. If user click on Delete action, "delete.php" page should delete the specific user from the database.