INET 2005 LAB 5

**Web Application Development I**

# Advanced Web Application Security

## Preliminary Steps

### Step 1: Have INET2005 Virtual Machine running.

### Step 2: Create a new PHP project in PhpStorm as we have been doing for this lab. Make sure the project is added to your Github repository.

### Step 3: Refer to the accompanying code samples for help as needed.

## Instructions

You will be adding additional security to your existing Assignment 1 application by doing the following:

### Step 1: Create a new MySQL Least-Privilege user account to replace the 'root' account that you are currently using and restrict the account to only allow execution of SELECT, INSERT, UPDATE, and DELETE of data, as well as the ability to execute Stored Procedures.

1. Ensure that your new account is not able to perform Data Definition Language statements such as DROP and ALTER.
2. You will show this account in MySql and demonstrate how you are connecting to the Employees DB by using this account instead of root

### Step 2: Create a new page in your application that allows the creation of new users.

1. The page could be called register.php. Make this page accessible along with the login page…you will of course have to make sure that the user doesn’t already exist and notify the user if the username is already in use.
2. Use the built-in bcrypt *password\_hash* function to ensure that your passwords are hashed before saving to your users database table.

### Step 3: Ensure your SQL statements are SQL Injection-Proof by replacing your current mysqli code with PDO prepared statements

1. Reference this W3 Schools article <https://www.w3schools.com/php/php_mysql_prepared_statements.asp>
2. You must replace all of your database calls for all functionality with prepared statements that will receive parameters instead of building SQL from scratch. In addition, one of your calls to the database, of your choosing, should be migrated to the database as a Stored Procedure and called from your code. Refer to the in-class lectures and this article (<http://php.net/manual/en/pdo.prepared-statements.php>) for further details.

### Step 4: Add a reCAPTCHA field to your login screen.

1. You can use the example code provided to implement your solution or feel free to research and add a different solution as there are many available. This article should provide you with good instruction (<https://www.kaplankomputing.com/blog/tutorials/recaptcha-php-demo-tutorial/> )-
2. Apply reCaptcha to your Login and Register pages.

NOTE: You can choose an alternate solution for this. Feel free to research and implement your own solution.

### Step 5: Move the entire Web Application to the Apache server on your Virtual Machine and apply a self-signed SSL Certificate to Apache so that your communications are SSL encrypted.

### 

1. Add a host entry to your VM's hosts file which will allow you to access your site via the address ***[yourname].dev .*** *(Example johndoe.dev). You can reference this article for help…*

<http://bencane.com/2013/10/29/managing-dns-locally-with-etchosts/>

1. Create the SSL certificate and configure the Apache web server on your VM by following the instructions in the following online article….

<https://www.digitalocean.com/community/tutorials/how-to-create-an-ssl-certificate-on-apache-for-centos-7>

* 1. You can ignore any sections in the article having to do with firewall configuration.

dept\_emp\_ibfk\_1

dept\_manager\_ibfk\_1

salaries\_ibfk\_1

titles\_ibfk\_1

select \* from employees join Employeeslastsalary using (emp\_no) where first\_name like concat('%emo%') or last\_name like concat('%emo%') order by Salary asc limit 100,25;

select emp\_no,MAX(salary) as Salary from salaries group by emp\_no;

select \* from Employeeslastsalary;