Project 1

The following assignment is worth 15% of your final overall grade.

The due date for your project code along with documentation is 12 noon December 6th. A document outlining how you performed the below tasks along with relevant source code snippets and test cases is to be provided. Your project code and documentation should be contained in a folder named as your student number “C001234XX” zipped and mailed to butlerr@itcarlow.ie.

**Create an authentication mechanism for a web application using XAMPP, PHP & MySQL. Your application should create the underlying database on requesting the login page as a root user with no password. NB: Please test this functionality before mailing me your code, as I cannot test your application without your underlying database.**

Your authentication mechanism should allow for the following functionality.

**Register with the system.** **(10% Total)**

* + The system should allow users to register with the system using a username and password. **Done**
  + Complexity rules regarding the password should be enforced. **Done**
  + Password storage should be salted and hashed. **Done**

**On an unsuccessful authentication attempt** **(20% Total)**

A generic error message is presented back to the end user outlining that the username & password combination cannot be authenticated at the moment. ie… “The username Richard and password could not be authenticated at the moment”. Note that the username supplied during the authentication attempt is reflected back to the user interface in the event of an unsuccessful login attempt.

* + Reflect the supplied username provided in the above message. Ensure that this reflected parameter in not susceptible to XSS. You are to write your own sanitisation code for characters that can be utilised for XSS.
  + Lockout after 5 attempts for 3 minutes. **DONE**

**On successful authentication** **(15% Total)**

* + The system should greet the user by their username. **Done**
  + Create an active authenticated session. **Done**
  + Allow for the authenticated user to view some pages (at least two) that an unauthenticated user will not have access to. **Done**
  + Allow for the user to logout securely. **Done**
  + Lockout after 10 minutes of inactivity. **Done**
  + Max session duration of one hour irrespective of in session activity. **Done**

**Password Change** **(15% Total)**

* + Authenticated users should be capable of changing their password. **Done**
  + Complexity rules regarding the password should be enforced. **Done**
  + On password change the active session should be expired. **Done**
  + The user will have to re-authenticate using new credentials to gain access to the system. **Done**
  + No out of band communication, mechanism is required to inform the user that their credentials has been updated. **Done**
  + You are to implement Cross Site Request Forgery (CSRF) protection on this page. **Done**
* Note: “In the real world” the values contained in this request would be passed as a POST request. **However, to expedite the correction & testing of this assignment you are to pass the values for this functionality in a HTTP GET request.**

**Event Log & ADMIN user (10% Total)**

* + Your application should store unsuccessful and successful login attempts to an event log. This event log should accessible and viewable to the authenticated user “ADMIN” only. **Done**
  + This users authentication details are as follows **Done**

Username = “ADMIN”

Password = “SAD\_2019!”

* + This account is to be created, when your database is being created. **Done**

**Testing**  **(30% for Test Cases and Results)**

Your documentation should include security test cases and test results for all implemented functionality. In this component of the report should highlight what security features you are assessing, the tests you performed along with the results.