Project

The following assignment is worth 40% of your final overall grade.

The due date for your project code along with documentation is **March 28th**. 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” and mailed to richard.butler@setu.ie.

**The Application contains numerous vulnerabilities that require the following to be documented for each:**

1. **The vulnerability type:**
2. **Steps to determine the existence of each vulnerability.**
3. **Your analysis of the vulnerability**
4. **Your mitigation strategy including code snippet**
5. **Your testing demonstrating that your mitigation works.**

Vulnerabilities in scope: **(I will Demo these in the Lab)**

* Reflective XSS (Your custom escaping function)
* Persistent XSS (Your custom escaping function)
* SQLi (Parameterised Query)
* CSRF (Hidden Field utilising CSRF Token)
* Brute force protection (This includes the full perimeter of the application)
* Directory Traversal (Allow access to appropriate application files only)
* Session Management Test Session management & Fix on all pages where appropriate
* Session Fixation (token change on successful login)
* Password Storage (Plane text, Implement Salt and Hash **NOT password\_hash())**
* Password Complexity (Enforce complexity on Registration)
* Page Caching (to be checked on all pages and tested on Firefox)
* Command Injection (Easy Fix - **escapeshellcmd()** )

The theory on items in red will be presented in class over the next two weeks.

**Important Notes:**

1. I will be testing your solution against on an identical machine to what is in L426. It is your responsibility to ensure your solution works on this setup.
2. I will be conducting interviews on your solution beginning the week March 31

You should be able to discuss each vulnerability, the impact of vulnerability on the application and the approach you have taken to mitigate the vulnerability.