System Requirements

Specification Index

For

Securing a Django Application: CSRF Protection and User Input Validation

(Topic:- Django Security Best Practices)

Version 1.0

Scenario: You are tasked with improving the security of a Django application that handles user registration and login. The application needs to ensure that:

- **CSRF protection is enabled to prevent cross-site request forgery attacks.**
- **User input is validated to avoid security vulnerabilities such as SQL injection, cross-site scripting (XSS), and other types of malicious input.**

Problem Statement: Your task is to:

- **Implement CSRF protection in the Django application, ensuring that all forms are protected from cross-site request forgery (CSRF) attacks.**
- **Implement user input validation in the form to ensure that the input is safe and secure before it is processed by the server.**
- **Test and verify that CSRF protection is in place and that the application can correctly validate user inputs.**

Tasks:

CSRF Protection:

- **Ensure CSRF protection is enabled globally in Django settings (CSRF_COOKIE_SECURE, CSRF_COOKIE_HTTPONLY, etc.).**
- **Ensure that every form in the application includes {% csrf_token %} in the template.**

User Input Validation:

- **Use Django's built-in form validation or custom validators to sanitize user inputs (e.g., for email, username, password fields).**

Execution Steps to Follow:

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to

   Application menu(Three horizontal lines at left top)->Terminal->NewTerminal.

3. The editor Auto Saves the code.
4. If you want to exit (logout) and to continue the coding later anytime(using Save & Exit option on Assessment LandingPage) then you need to use CTRL+Shift+B command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while

   logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.

6. To test any Restful application, the last option on the left panel of IDE, you can find

   ThunderClient, which is the lightweight equivalent of POSTMAN.

7. To test any UI based application the second last option on the left panel of IDE, you can

   find Browser Preview, where you can launch the application.

8. Install ‘djangorestframework’ module before running the code. For this use the following command.

   pip install djangorestframework

9. Use the following command to run the server

   python3 manage.py runserver

10. Mandatory: Before final submission run the following commands to execute testcases

    python3 manage.py test library.test.test_functional

    python3 manage.py test library.test.test_exceptional

    python3 manage.py test library.test.test_boundary

11. To test rest end points

    Click on ‘Thunder Client’ or use Ctrl+Shift+R->Click on ‘New Request’ (at left side of IDE)

12. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on “Submit Assessment” after you are done with code.

13. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.