System Requirements

Specification Index

For

Implementing Custom Django Middleware for Logging Requests and Responses

(Topic:- Django Middleware)

Version 1.0

Scenario: As a Django developer at a growing tech company, you have been tasked with improving the observability of the application. The company needs to monitor and log every incoming HTTP request's URL and its corresponding response status code. This information will help in tracking the usage of different endpoints and debugging potential issues.

Your task is to write a custom Django middleware that logs the request URL and the response status code to a log file.

Create a Django middleware that logs the following:

- **The URL of the incoming request.**
- **The status code of the response after the request is processed.**
- **Ensure the log entries are saved with timestamp information.**
- **The middleware should be configurable (i.e., it should be possible to enable or disable logging easily).**
- **Log the data to a file named requests.log.**

Your Task

- **Write a custom middleware that logs the request URL and the response status code.**
- **Configure the middleware to write logs into a file named requests.log in the logs directory.**
- **Make sure to handle both successful requests and errors by logging the response status appropriately.**
- **Test the middleware to ensure the correct URL and status codes are being logged.**

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.