System Requirements

Specification Index

For

Managing User Data in Django: Implementing a Command to Delete Inactive Users

(Topic:- Django Custom Commands)

Version 1.0

Problem Statement Description:

Scenario:

You are working as a Django developer for a project that manages user accounts. The application needs a management command to handle cleaning up inactive users in the database. This is especially important for keeping the database clean and maintaining optimal performance by removing inactive users who have not logged in for a specified period.

Problem Statement:

Your task is to create a Django management command that deletes all inactive users from the database. An inactive user is defined as a user who hasn't logged in within the last 6 months. The command should handle potential issues like invalid data, and the system should not crash if there are no inactive users to delete.

Your Task:

- **Create a management command that deletes inactive users.**
- **Define an "inactive user" as one who hasn't logged in in the past 6 months.**
- **Ensure proper exception handling and validation to prevent issues during deletion.**
- **Test the command to ensure it deletes the correct users.**

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.