System Requirements

Specification Index

For

Database Query Optimization in Django

(Topic:- Django Performance Optimization)

Version 1.0

Scenario: You are working as a Django developer for a startup that provides an online platform for event management. The platform allows users to browse and register for events. As the platform has grown, users are reporting slow page loads, especially when browsing events with many participants and interactions. The issue is caused by inefficient database queries.

Your task is to optimize the database queries for the event listing page, which shows a list of events with details like event name, date, and the number of participants. Currently, the database queries are unoptimized, leading to unnecessary database hits, and the page is slow to load.

Problem Statement

You need to optimize the queries for the event listing page. Each event includes the event name, event date, and the list of participants. Optimizing the queries will help in reducing the number of database hits and improve the performance of the page.

Your task:

- **Identify the problem: Detect the N+1 query problem when querying events and their related participants.**
- **Optimize the query: Use Django ORM techniques such as select_related and prefetch_related to optimize the query.**
- **Verify the solution: Ensure that the queries are optimized and test the performance improvement.**
- **Write test cases to verify that only the necessary queries are made and the optimization is correct.**

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.