Date: 03-12-2020

Course: TY BCA 5th Semester

Enrollment No.: 201803100110002

Subject: 030010517 CC12 Software Testing Techniques

Set: A

Worksheet 8

a. Configure the Performance Test environment for the registration and login page of the site.

Answer:

Steps for the test to webservice

Step 1: Download and install JMeter. Then open JMeter

- Open the JMeter window by clicking on D: JMeter -> bin -> jmeter.bat

Step 2: Create a JMeter test plan

- In the left side you can see the elements of the test plan. Right side contains all the configurations of that element.
- Test plan contains all the elements that is need to perform your test and Workbench can keep your elements for a temporary basis. You can provide any name to your JMeter Test.

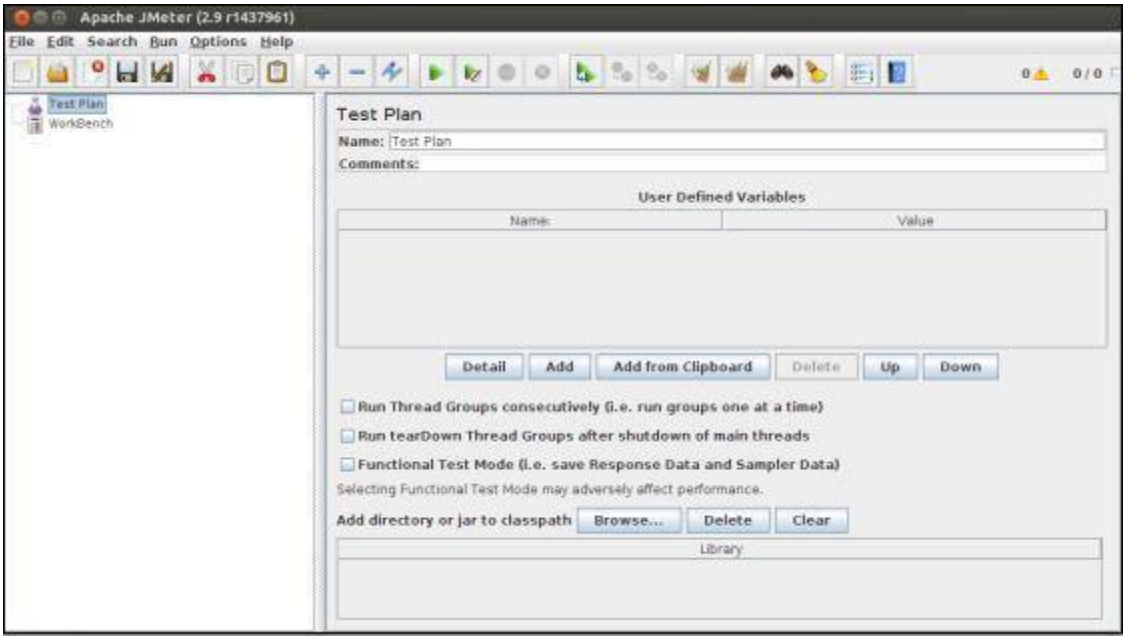


Figure 1 Create a test plan

Step 3: Create thread group

- If you right click your Test Plan and go to Add, you can see various elements under a drop-down list. Some of them are Threads (Users), Test Fragments and Config Element etc.
- In JMeter, all the node elements are added by using the context menu.
    - Right-click the element where you want to add a child element node.
    - Choose the appropriate option to add.
    - Right-click on the Workbook 8 (our Test Plan) → Add → Threads (Users) → Thread Group. Thus, the Thread Group gets added under the Test Plan node
    - Name the Thread Group as Users.

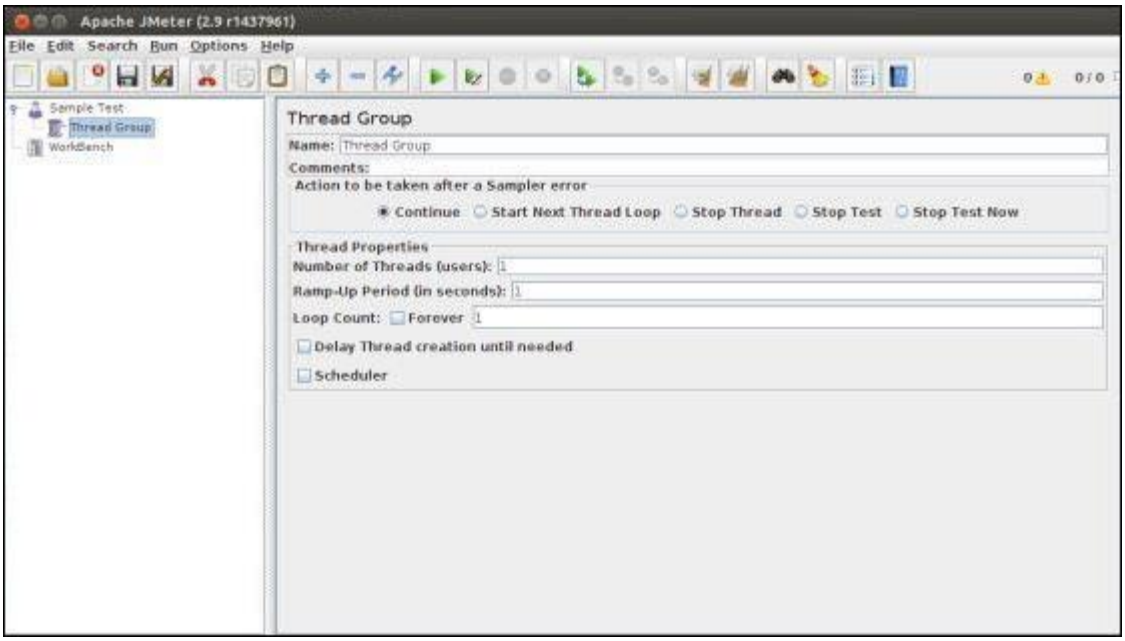


Figure 2 Create thread group

Step 3: Add Sampler

- We need to add one Sampler in our Thread Group (Users). As done earlier for adding Thread group, this time we will open the context menu of the Thread Group (Users) node by right-clicking and we will add HTTP Request Sampler by choosing Add → Sampler → HTTP request option.

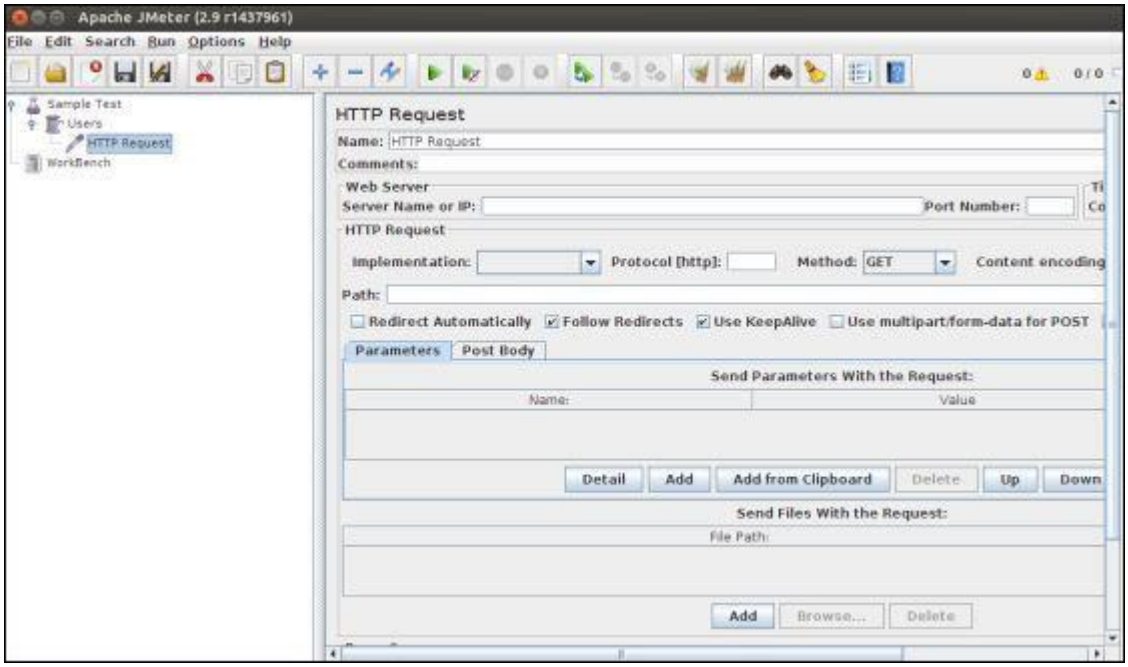


Figure 3 Add Sampler

- It will add one empty HTTP Request Sampler under the Thread Group (Users) node. Let us configure this node element.

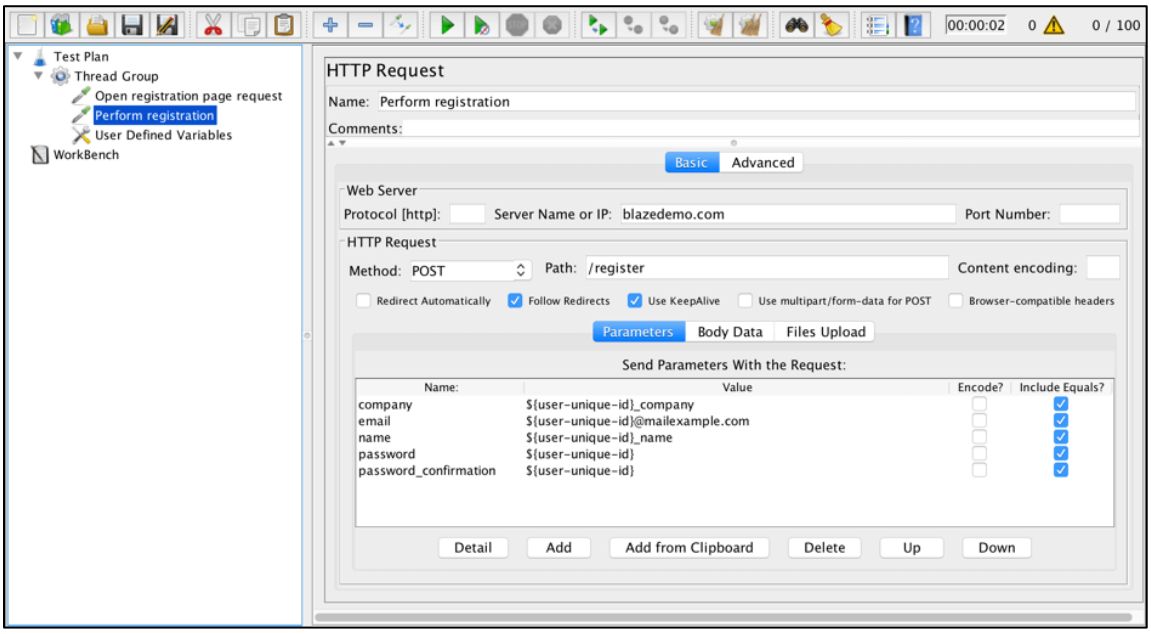


Figure 4 Add Sampler for registration page

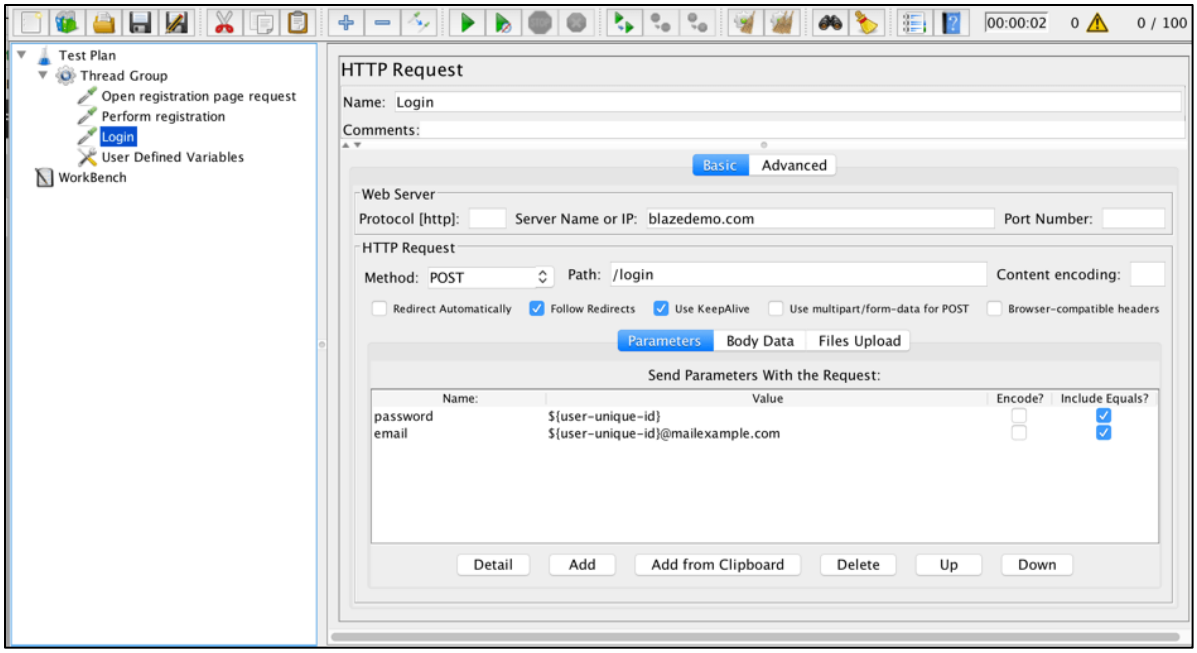


Figure 5 Add Sampler for login page

- **Name** – We will change the name to reflect the action what we want to achieve. We will name it as Visit Maharashtra Tourism Home Page
- **Server Name or IP** – Here, we have to type the web server name. In our case it is www.tutorialspoint.com. (http:// part is not written this is only the name of the server or its IP)
  - **IP: mtdcrrs.maharashtratourism.gov.in**
- **Protocol** – We will keep this blank, which means we want HTTP as the protocol.
- **Path** – We will type path as / (slash). It means we want the root page of the server.
  - For Registration page: /Registration.aspx
  - For Login Page: /Login.aspx

Step 4: Add Listener

- We will now add a listener. Let us add View Results Tree Listener under the Thread Group (User) node. It will ensure that the results of the Sampler will be available to view in this Listener node element.
- To add a listener
  - Open the context menu
  - Right-click the Thread Group (Users)
  - Choose Add → Listener → View Results Tree option

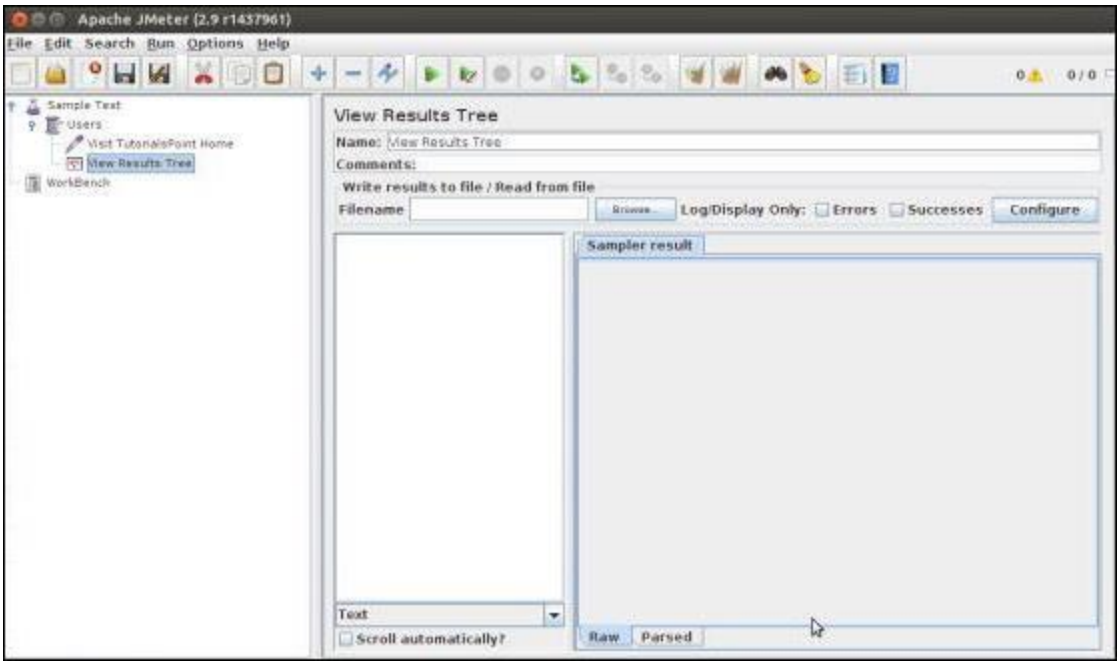
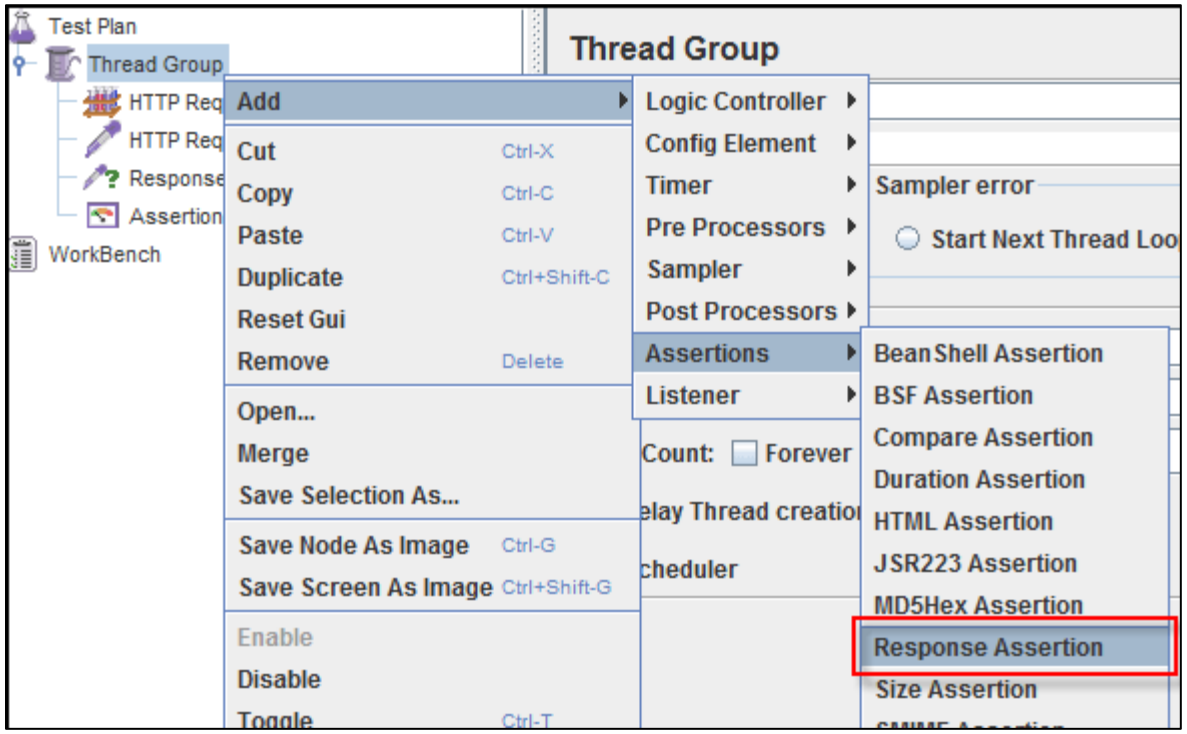


Figure 6 Add Listener

b. Implement following assertions and retrieve the results:

i. Response assertion



- Let's see how to add an assertion, go to your Thread Group, in my case it is Users. Right click your Thread Group, go to Assertions and click on Response Assertions.
- You can apply an assertion to the Main sample and sub samples, Main sample only, Sub-samples only or apply to JMeter variable.
- The next field in the Response Assertion window is the Field to Test. In my case I have clicked Response Code as the field to test my response. Also, you can test Text Response, Document (text), URL sampled and so on.
- I want to check whether the response code is Equals 200, so for that we can write the code in the Patterns to Test text box. This assertion will check the response code and compare with 200, if this matches it will pass the test case or if not, it fails.

ii. Listener: Assertion results

- It is used to view the result of assertions.

iii. Duration assertion

- Checking the duration of the response.
- We can apply it to Main sample and sub samples, Main sample only or Sub-samples only**.**
- Here I have added 5000 ms as the Duration. If the request takes more than 5000 ms will be a failure and will get notified in the Assertion Results.

iv. Size assertion

- As name indicates it is related to the response size. In my test I have added Size in byte as 160640 and I have added Type of comparison as greater than equal sign (>=).
- If the request is greater than equal to 160640, then it should pass else it should fail. Also, you can use multiple comparisons available under Type of comparison.

v. HTML assertion

- Test the format of the response. It checks whether the format is valid HTML or not. If it is not a valid HTML it will throw some errors and fail the assertion.

vi. XML assertion

- This will check the format, if it is a valid XML it will pass otherwise it will fail.
- XML Assertion failed and it says that the mark-up in the document preceding the root element must be well-formed.

c. Add listeners

i. View results in table

- It contains following parameters**.**
    1. **Samples**: Number of samples
    2. **Start Time**: Start time of the request
    3. **Thread Name**: Users (Name of your thread group)**,**
    4. **Label**: Name that you have provided to HTTP request sampler
    5. **Status**: Status is based on the assertions in the test, if assertion(s) passed then status is successful with green colour icon, otherwise it will show as red.
    6. **Bytes**: Size of the request

7. **Latency**: is a difference between time when request was sent and time when response has started to be received
8. **Connect Time (ms)**: The time request took to connect to the server.

ii. View results in tree

- We can see the results in tree structures.
- We can get some information like response code, response message, some headers and some other information**.**

iii. Aggregate report

- In view Result Tree and view Result Table display one line per request but in aggregate report it provides single line of response.
- The advantage that is available in aggregate report is that it accumulates all the metrics and provide single line of report.
- It contains following information
    1. **#Samples:** number of samples
    2. **Average**: average time taken by all the samples to execute specific label
    3. **Median**: half of the request took more than this 952 ms and the half of the request took less time than this particular number
    4. **90% Line**: 90% of the request took less than 1007 ms or 90% request are within this range
    5. **95% Line:** 95% of the request took less than 1028 ms or 95% request are within this range
    6. **Min**: minimum time request took to execute
    7. **Max**: maximum time request took to execute
    8. **Error %**: error percentage
    9. **Throughput**: number of requests per second
    10. **Received KB/sec**: received response size per second
    11. **Sent KB/sec**: sent request size per second

iv. Graph results

- We can see dots in the graph. Green coloured dots will be the Throughput, red coloured ones are Deviations, purple colour is for Median and blue colour dots for Average and Data is in black.

v. Summary report

- The result of the summery report is similar to the aggregate report.

vi. Simple data writer

- This listener you don't have any UI, you just have a box where we can browse our file (file you need to log your result).

- **Enhancement Learning**
  - ✓ **What is Post-Deployment Test?**

    Answer:
    - Post deployment testing is a concentrated period of testing that follows the deployment of a website or app into the live environment.
    - A variety of compatibility and exploratory checks identify any deployment or configuration issues in the live environment which may have been obscured in test environments

  - ✓ **List two parameters of Database Testing.**

    Answer:

    The below parameters are tested during Database Testing**.**
    - Data Integrity
    - Store Procedures
    - Triggers
    - Field validations
    - Constraints

  - ✓ **Describe preparation of Database Test Plan in JMeter.**

    Answer:

    ⇒ Following steps are to be followed during Database Test plan in JMeter.

    ⇒ Create a JMeter Test Plan
    - Go to your JMeter bin folder and double click on the ApacheJMeter.jar file to launch JMeter interface.
    - Select the test plan node and right click on the selected item.
    - Mouse hover on "Add" option, then elements list will be displayed.
    - Select Threads (Users) > Thread Group.
    - Rename this thread group as JDBC users.
    - The default properties of the thread group remain unaltered.

    ⇒ Adding JDBC requests
    - Right click on the JDBC users element.
    - Mouse hover on "Add" option, then elements list will be displayed.
    - Select Config Element > JDBC Connection Configuration.

    ⇒ You need to set up some important fields which will determine the proper connection between your database and JMeter. These fields include -
    - Variable name bound to pool - It identifies the configuration uniquely. This name will further be used by the JDBC Sampler to identify the configuration to be used. We have named it as test.

- Database URL - jdbc: mysql://localhost:3306/mywebsite
- JDBC Driver class - com.mysql.jdbc.Driver.
- User Name - root.
- Password - password for root.
- Other fields are left unaltered.
- Right click on the JDBC users element.
- Mouse hover on "Add" option, then elements list will be displayed.
- Select Sampler > JDBC request.
- Select this new element to view its control panel.

⇒ You need to set up some important fields. The JDBC request control panel includes

- Name- JMeter.
- Enter the Pool Name - test (same as in the configuration element).
- Query Type - Select statement.
- Query- Enter the SQL Query string field.