**JMeter Results Analyzing Center**

**Introduction:**

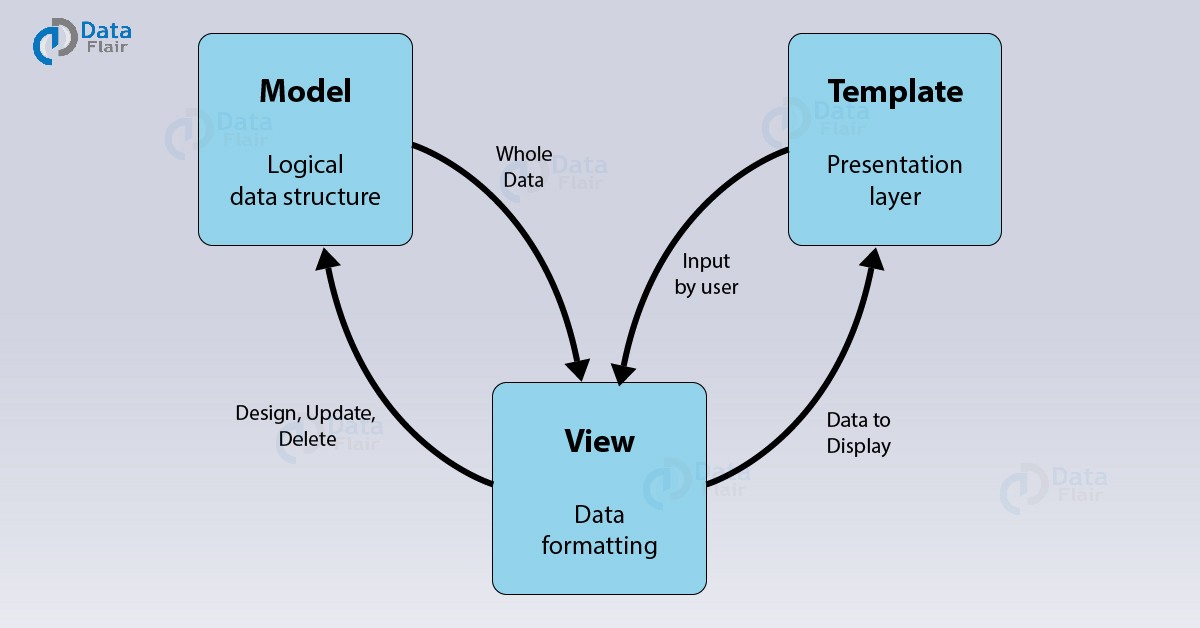
JMeter results analyzing Center is a platform where we can able to do a comparative analysis between different load tests provided and download the customized report by uploading the JTL files as well. It can avoid the excel operations like VLOOKUP’s, Milliseconds to Seconds Conversions, Error Messages, Error Codes and Individual request comparison (last 5 tests) etc …

Online web-application/dashboard for Load testing with JMeter. A central system for launching Load Test, monitoring tests, Creating & Storing reports and for a comparative analysis between different load tests provided with JMeter. Can be used with Jenkins or as a replacement for Jenkins + Plugins + JMeter combination for Continuous Integration.

**Technology Used:**

* Language: Python
* Template: Django
* Technique: Code First
* DB: PostgreSQL
* Editor: PyCharm

**Architecture:**



**Django Admin:**

* Django admin have right to create, delete and update Projects Names and their parameters.
* JMeter Tests can also be configured from admin by providing required parameters like Path, Project, Start Time, End Time and Started by etc.…
* Users and Groups are registered by admin only.
* Admin have a right to provide projects view access to specific users only.

**Admin Credentials:**

URL: localhost:8989/admin

Username: PTADMIN

Password: Password\_1

**Modules:**

* Analyzer - build reports, analyze results and compare results with another.
* Online - online monitoring for running tests (Needs to implement)
* Controller - configure and run the tests (Needs to implement)
* Administrator - configure different parameters

**Pre- Requisites:**

* **Postgres SQL – Version 12.0**
* **Python – Version 3.7**
* **Upgrade PIP**
* **Django – 2.2.6**
* **NumPy**
* **Pandas**
* **Matplotlib**
* **SciPy**
* **SQLAlchemy**
* **Psycopg2**
* **Para Miko**
* **Psutil**
* **WFastCGI**
* **Mathfilters**
* **Xlwt**

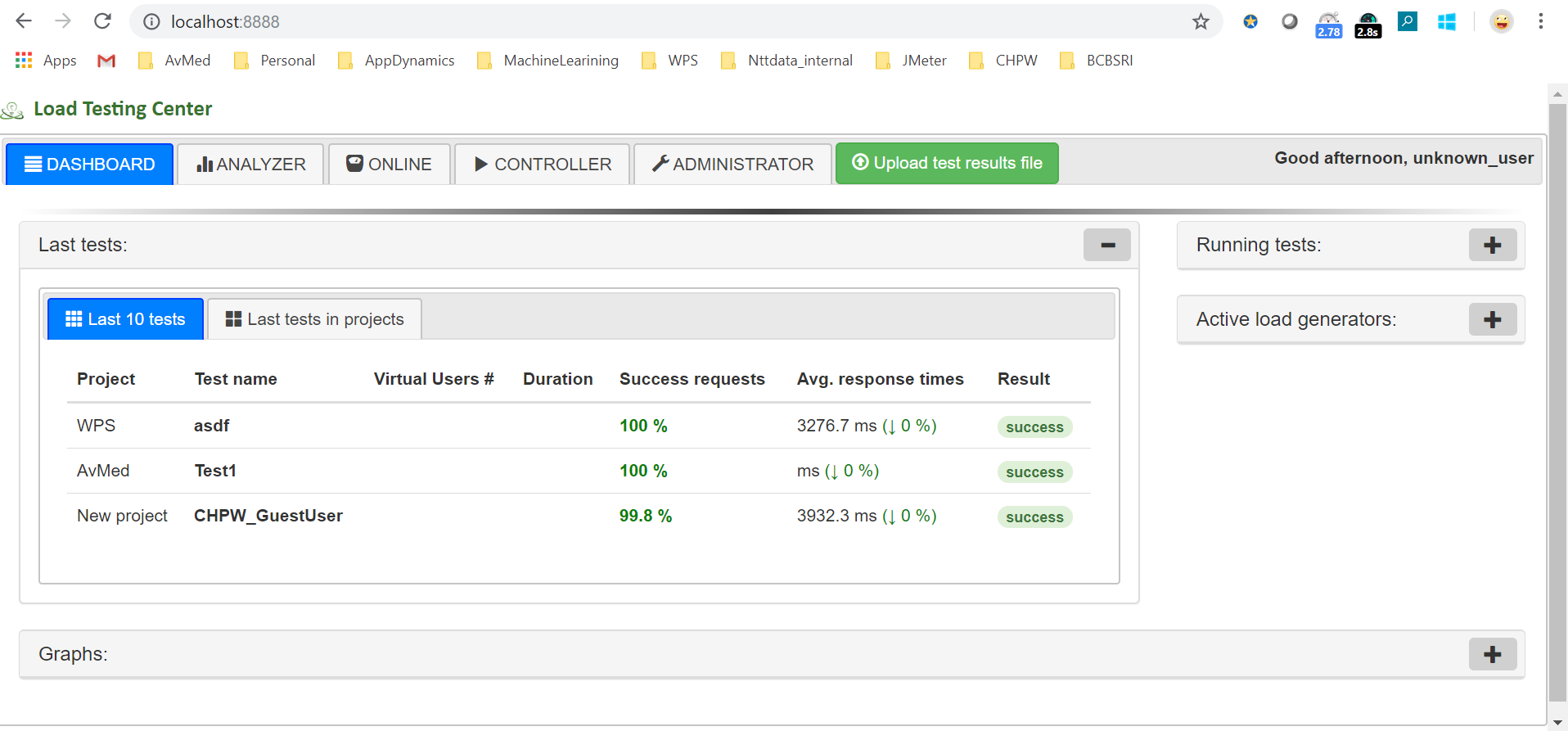
**Summary:**

After integration of Postgres SQL, Project name should be created in **Controller** tab then JTL files needs to be uploaded from **Upload Test Results File** tab. The uploaded test results data will be saved in Postgres SQL database. Now we can analyze the uploaded test results in **Analyzer** tab which is fetched from DB by selecting the project name in the dropdown. We can compare the current test results with previous test results within the same project and export the comparison results to Excel. We have an **Additional** Section which provides missing Requests/Transactions information from Comparison results. Along with the above results we can able to generate and export Aggregate Table, Response time Graphs and Composite graph results.

Online and Controller are the modules which is used to run the JMeter test and able to monitor the test results like Response Times, Error Percentage, Error Messages etc... while test is running. It will help us to find out the issues at early stages.

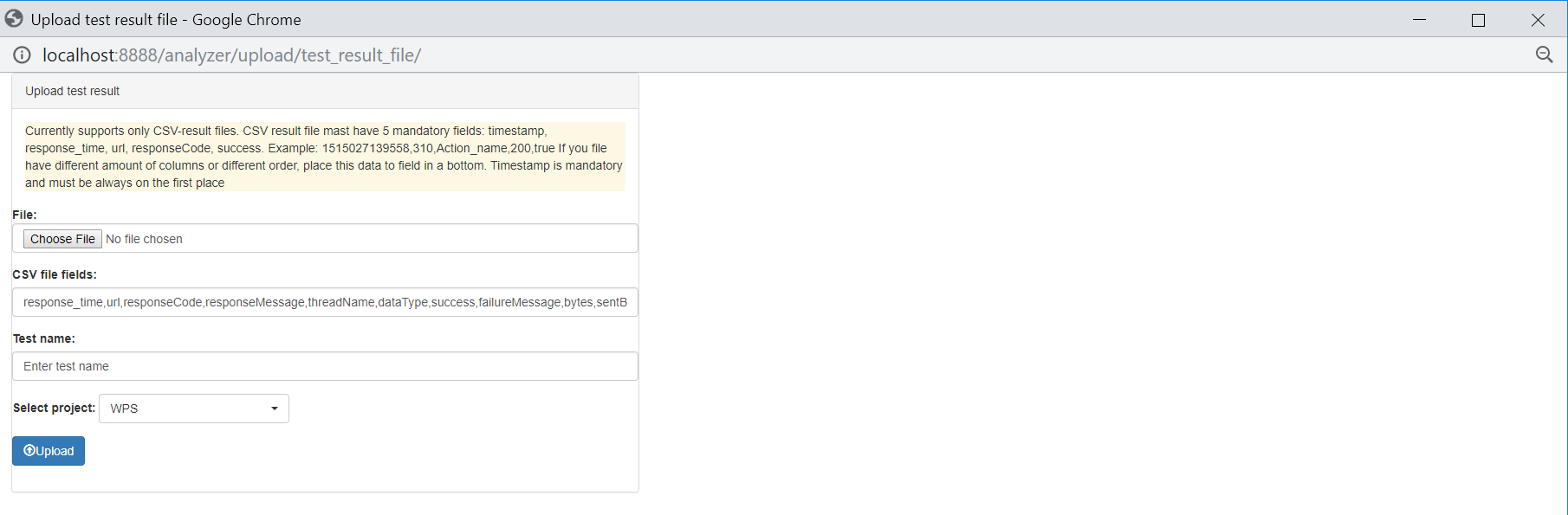
**Dashboard:**

Dashboard contains last 10 tests details like Project Name, Test Name, Virtual users, Duration, Success% etc.…



**Upload Test Results File:**

We can Upload the JTL files in this section.

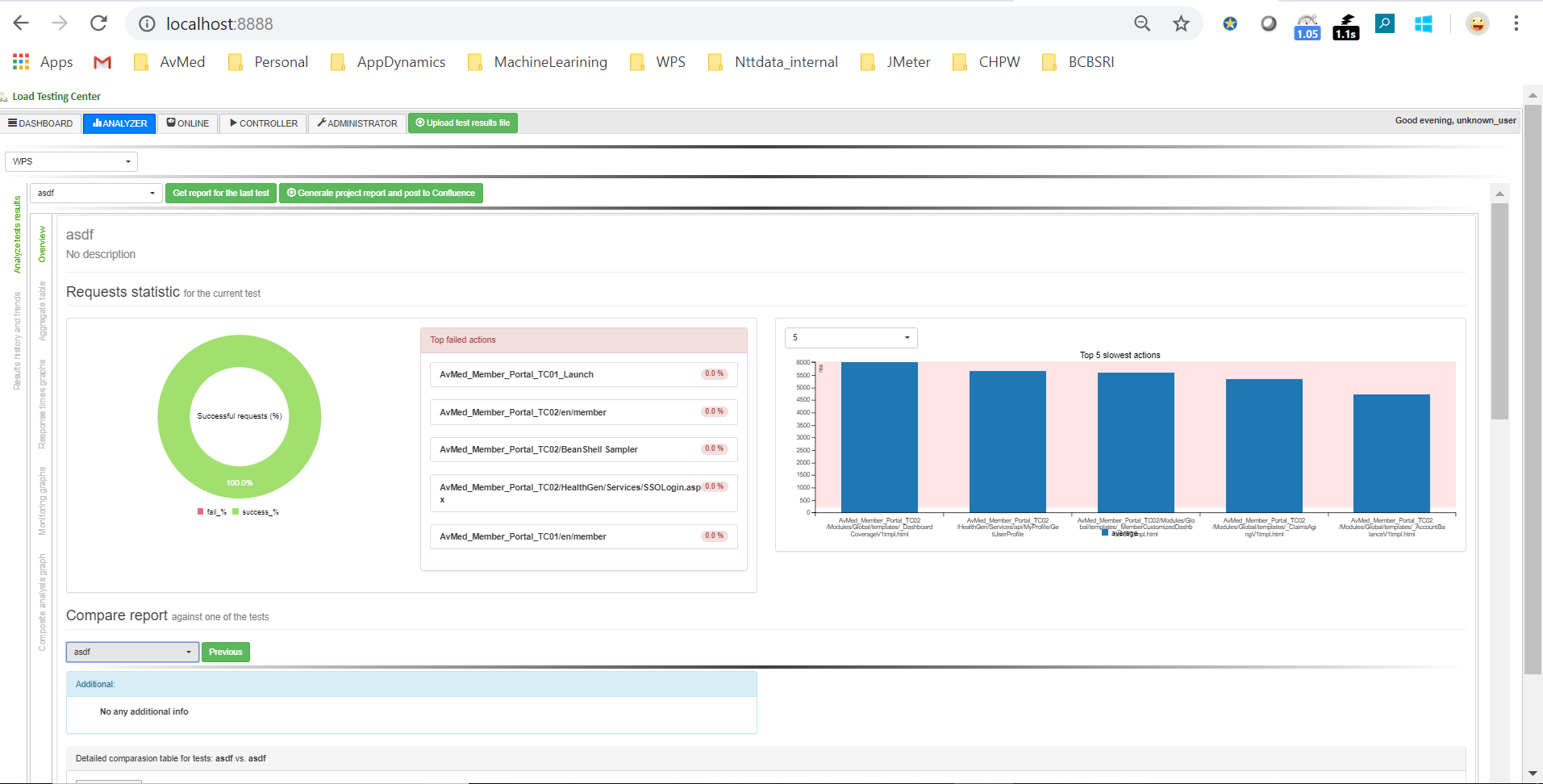


**Module 1 - Analyzer:**

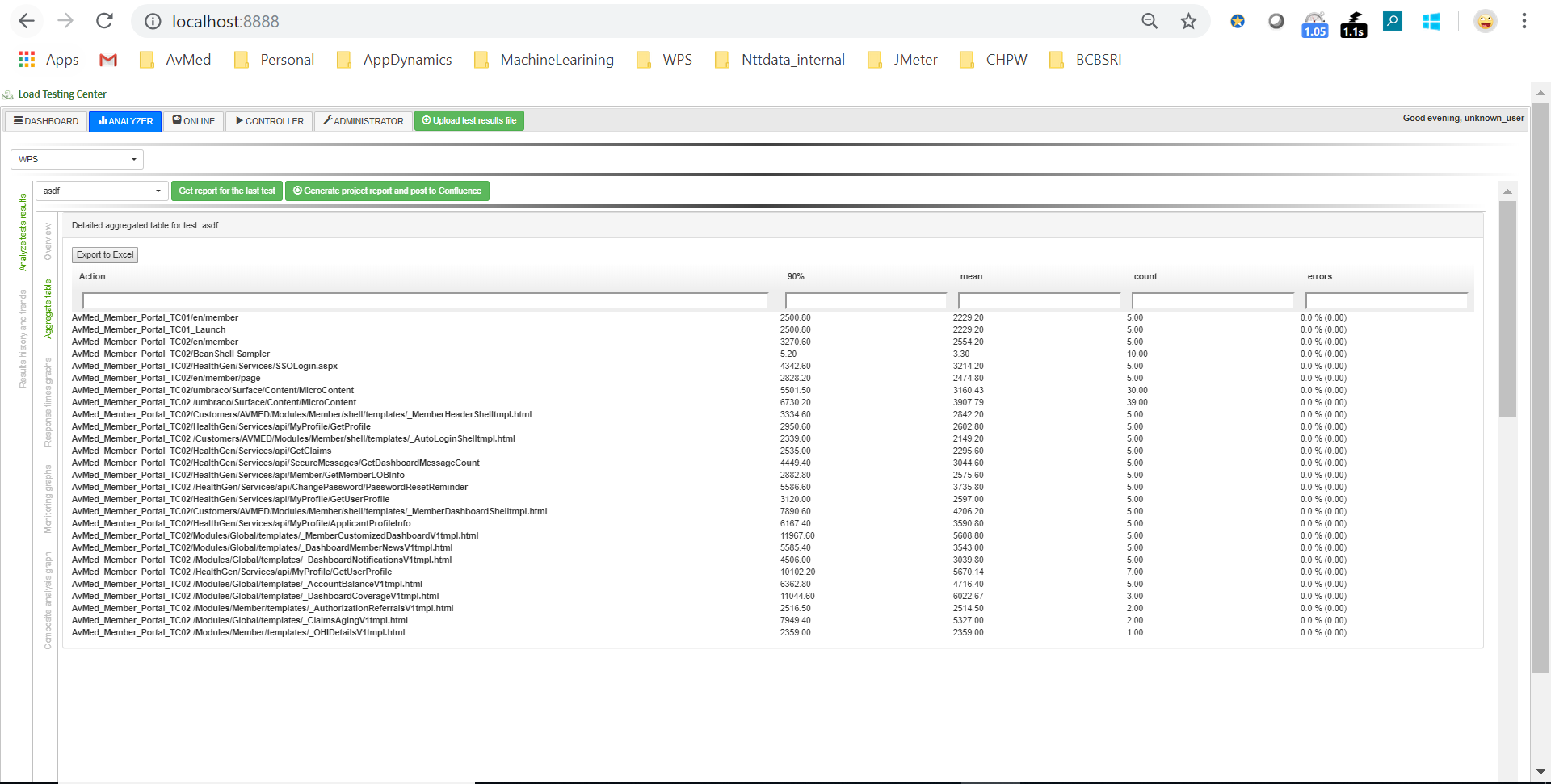
In Analyzer dynamic report for the tests can be generated, comparison of current and previous test results can be done, Aggregate report, Response time graph and composite analysis graph will be generated for current test. In the Aggregate Report ,Error details is also visible at the bottom. Individual requests can be monitored along with Boxplot, Error Details and Average/median response times (Milli Seconds).

**Sub Modules in Analyzer:**

* Overview
* Graphical view of Success and Failed Percentage
* Top Failed Requests
* Top slowest requests can be filtered
* Compare load test reports(Any 2 tests)
* Generate Excel Report
* Aggregate Table
* View Aggregate Table and Export to Excel
* Individual requests can be drilled down to monitor Average/Median response times and errors during the Test
* Boxplot contains Min, Max, Mean, Median etc.…
* Response Time Graphs
* View Response Time Graph with a Granularity - 1Min
* Able to view Individual Request Response times
* Monitoring Graphs (Needs to be Implemented)
* Composite Analysis Graph
* Response times comparison graph of individual requests of any 2 tests.

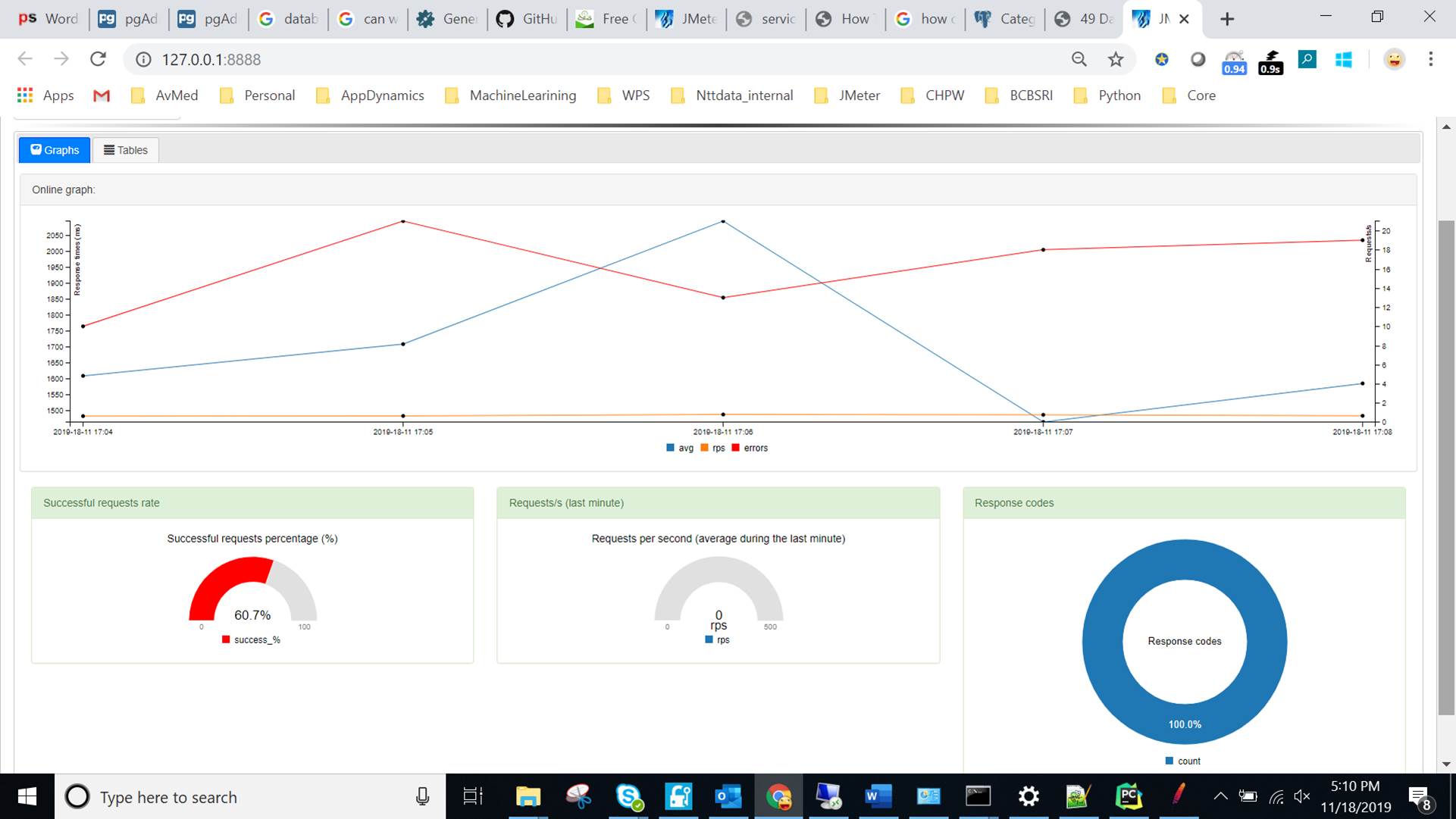


The aggregate table will look likes as below



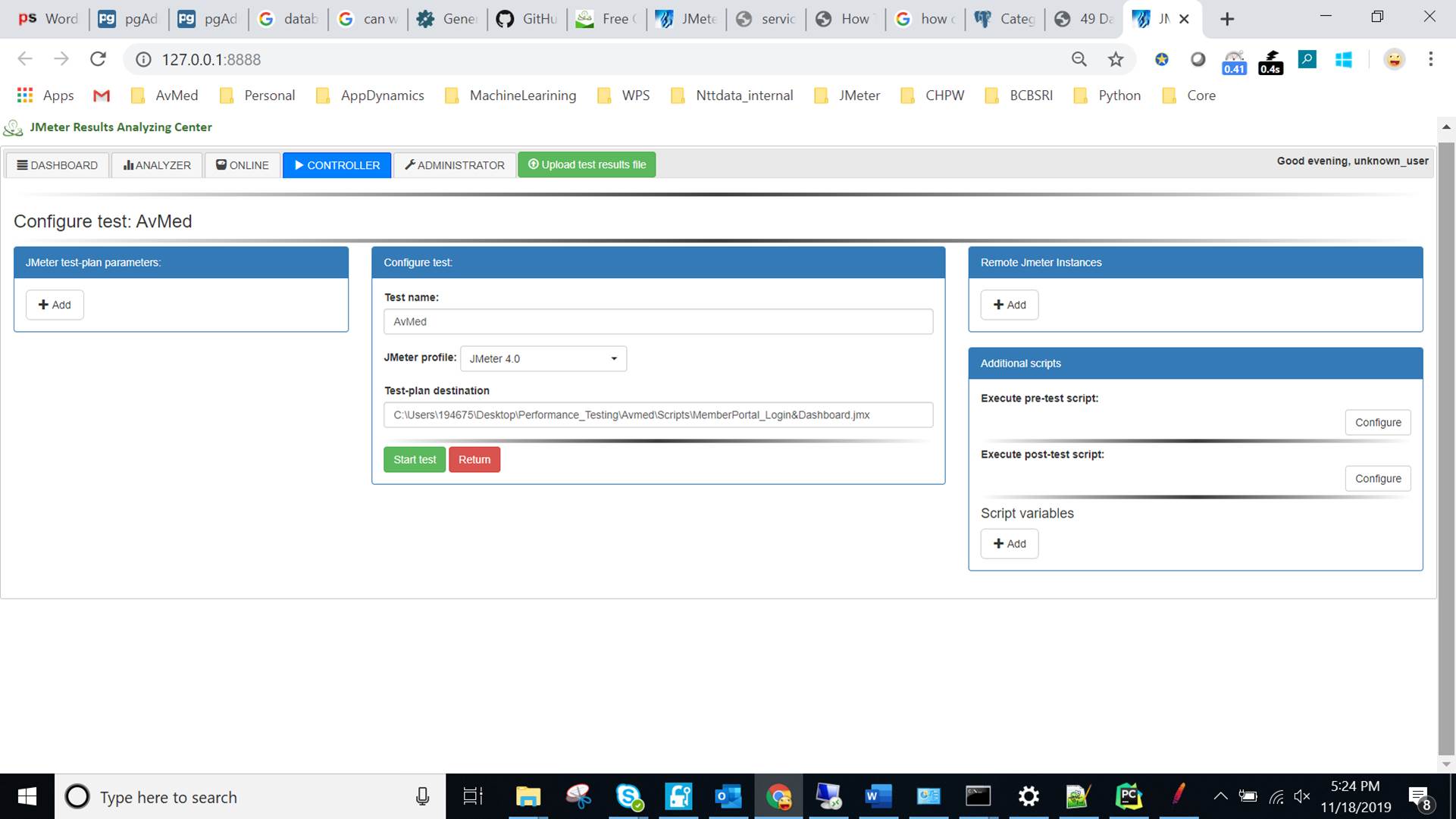
**Module 2 – Online :**

Able to monitor the test parameters like Average Response times, Success Rate, Request/Sec, Response Codes and Aggregate table while test is running. This can be achieved only when we run the test through Controller. Results are updated in database automatically while the test is running.



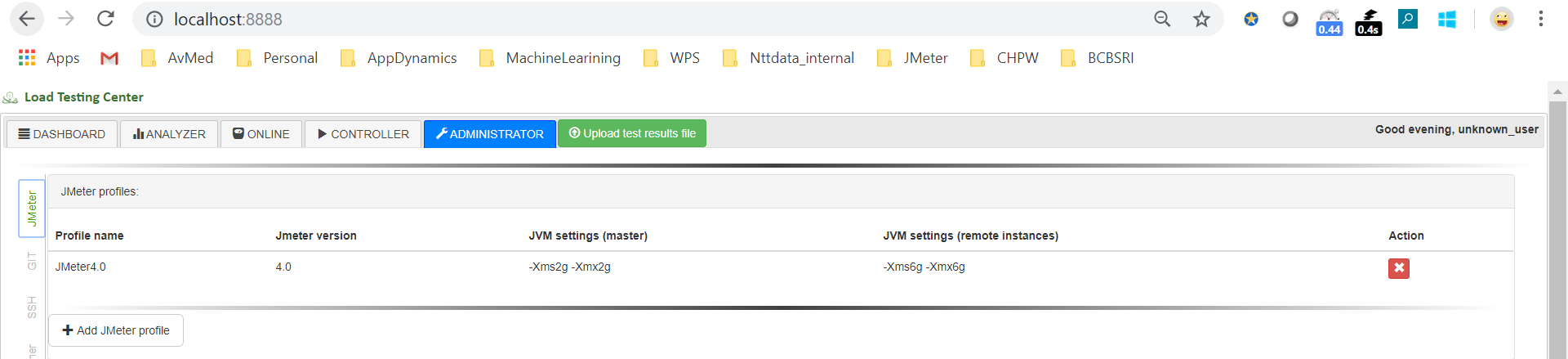
**Module 3 – Controller :**

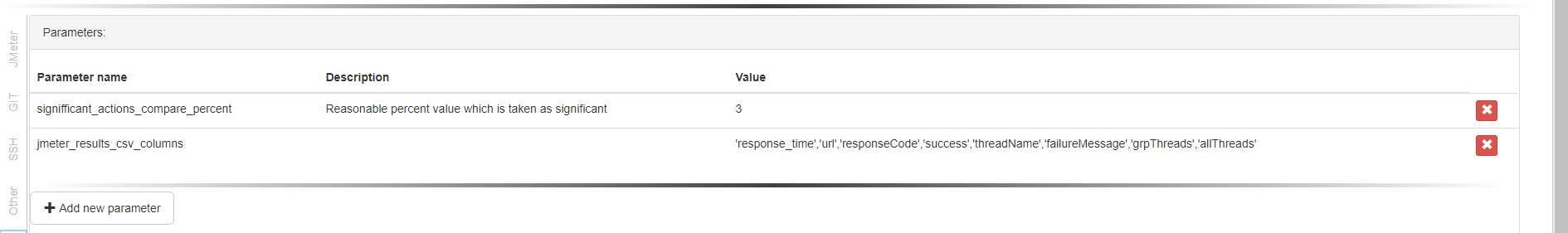
Controller is used for Creating new project, Running tests and Scheduling the tests. Configure JMeter test plan parameters like Users, Duration etc.… and test parameters like Project Name, JMeter destination, Script destination, add LG instances and Jenkins integration like Pre-Test script executions and Post-Test script executions. For Online Monitoring JMeter test needs to be executed from here.



**Module 4 – Administrator:**

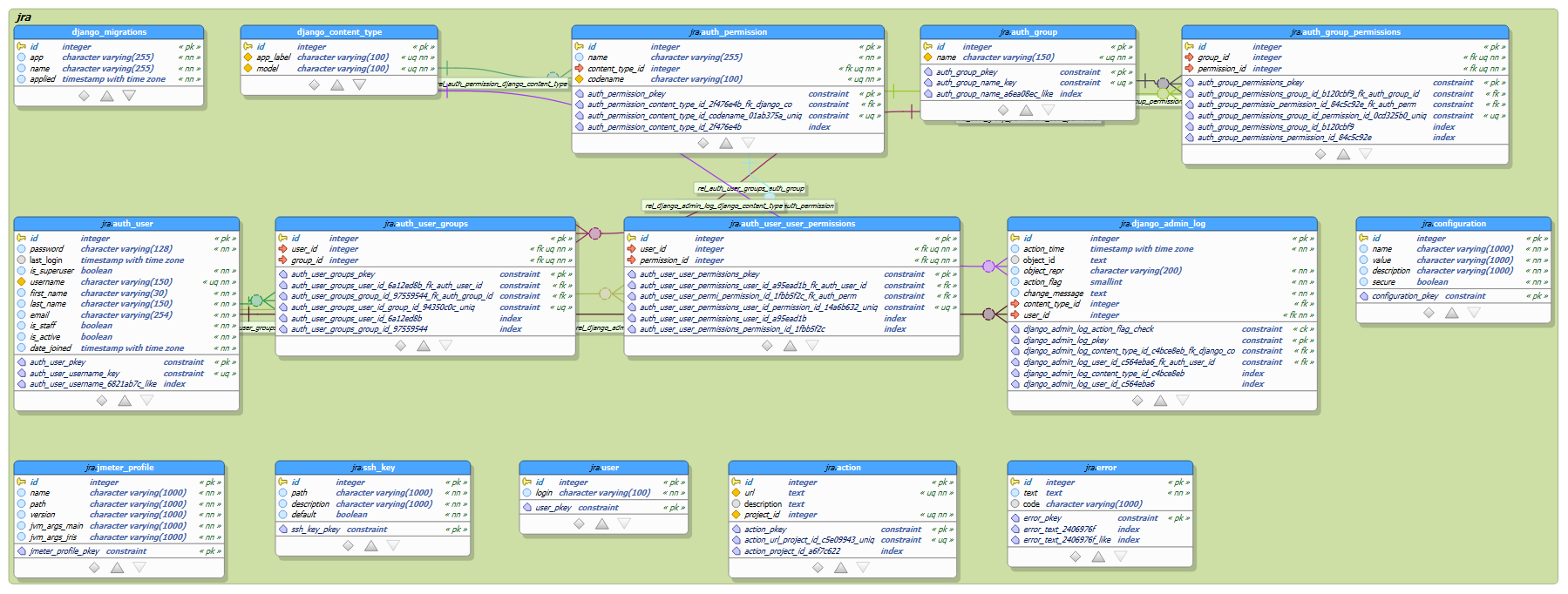
Configure JTL files parameters related to upload test results file and JMeter Heap memory Settings.





**Database ER Diagram :**

The below diagram represents the entity relationship between the database tables



* While running the test, results will be saved in staging tables (temporary tables) and it will be moved to actual tables once test got stopped.
* For every one minute the data will be refreshed in database while test is running , same is refreshed in UI(Online).
* User sessions, JMeter configs, Profile settings, Java Path, Auth related data and LG’s info are stored in appropriate tables.
* Totally 37 Tables are holding the data.

**DB Info: POSTGRESQL**

------------------------------

* Database Name: JRAC
* Schema Name: jra
* Username: postgres
* Password: Dilli1347

**Installation Process:**

1. Install python 3.7
2. Upgrade the pip
3. Install PostgreSQL 12.0
4. Create data base and schema as JRAC and jra
5. Create virtual env (python -m venv path)
6. Activate virtual env (run activate.bat file in env.scripts)
7. Run the listed requirements which are there in the requirementstoinstall.txt (pip install -r requirementstoinstall.txt)
8. Make migrations (python manage.py makemigrations)
9. Migrate (python manage.py migrate) it will create database tables.
10. Load initial data through fixtures/initial\_data.json
11. Then run server or run though command from local python manage.py runserver 8888(user defined port)

**Sample Reports:**

Refer the below attached sample reports downloaded from JRAC.

**Conclusion/Benefits:**

Using this have many benefits mentioned below which saves human effort and accuracy.

* **Time Saving**
* **Customized Reports**
* **Comparison of 2 Tests and Improvement % Details**
* **Export the results into Excel**
* **Identify the Issues at Early Stages**
* **Excluding the Excel Operations like VLOOKUPS, Seconds Conversions**
* **Data Storage in DB**
* **Easy of running JMeter Executions**