Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Верификация программного обеспечения

**ОТЧЕТ**

к лабораторной работе № 7

Выполнил:

студент гр. 153503 Кахновский Е.С.

Проверил:

Чайкин И.С.

Минск 2023

В качестве сайта для проверки я выбрал сайт собственного написания (лабораторная работа по учебной дисциплине СТРweb-ПР) (рисунок 1). Мой выбор был основан на необходимости тестирования данного сайта, так как ранее он не был тестирован.



Рисунок 1 – Главная страница сайта.

Для данной лабораторной работы был использован браузер Mozilla Firefox, а также установлен JMeter — инструмент нагрузочного тестирования с возможностью создания большого количества запросов одновременно. Все нагрузочные тесты были созданы с помощью данного инструмента.

В ветку Test Plan был добавлен элемент Thread Group и выставлены значение некоторых настроек (рисунок 2):

* Number of Threads (значение – 1000): количество имитируемых пользователей, одновременно работающих с сайтом;
* Ramp-up Period (значение – 3): промежуток времени, через который выполняется запуск очередного процесса;
* Loop Count (значение – 5): количество раз, которое будет выполняться сценарий внутри Thread Group;
* Action to be taken after a Sample Error (значение – “Continue”): действие, выполняемое в случае, если запрос вызовет ошибку.

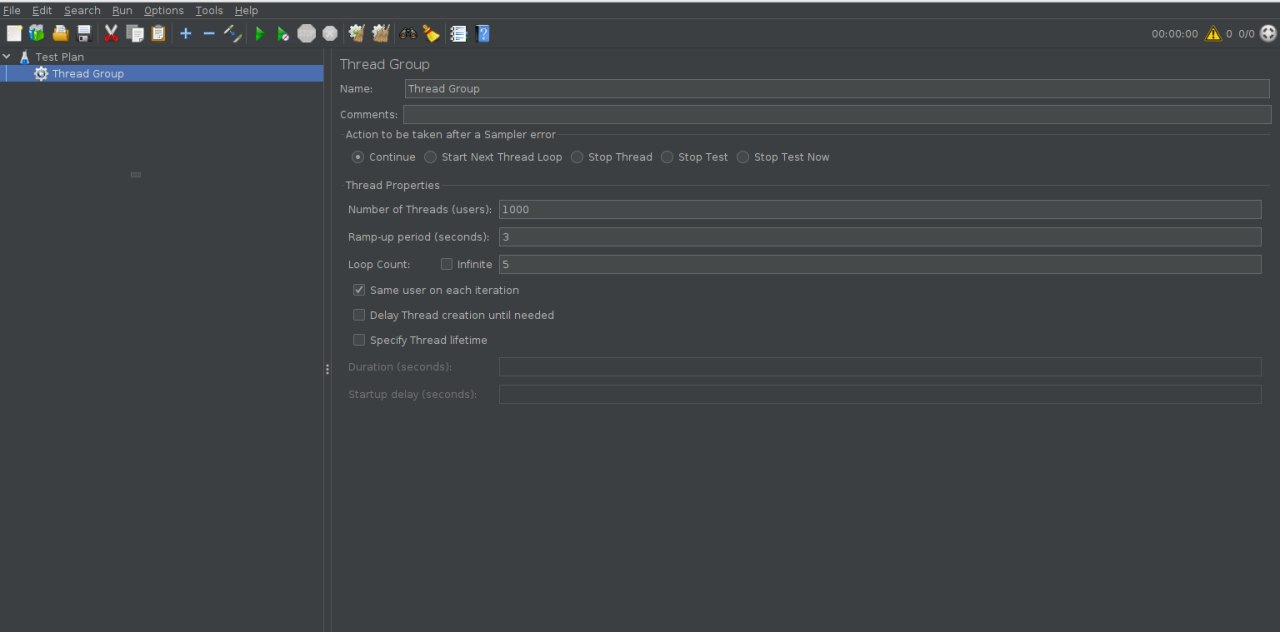


Рисунок 2 – Элемент Thread Group.

Далее был создан элемент HTTP Request, который является сценарием теста, т.е. набором различных действий для создания нагрузки на сайт. В окне настроек установлены адрес сайта, порт и путь к странице (рисунок 3).

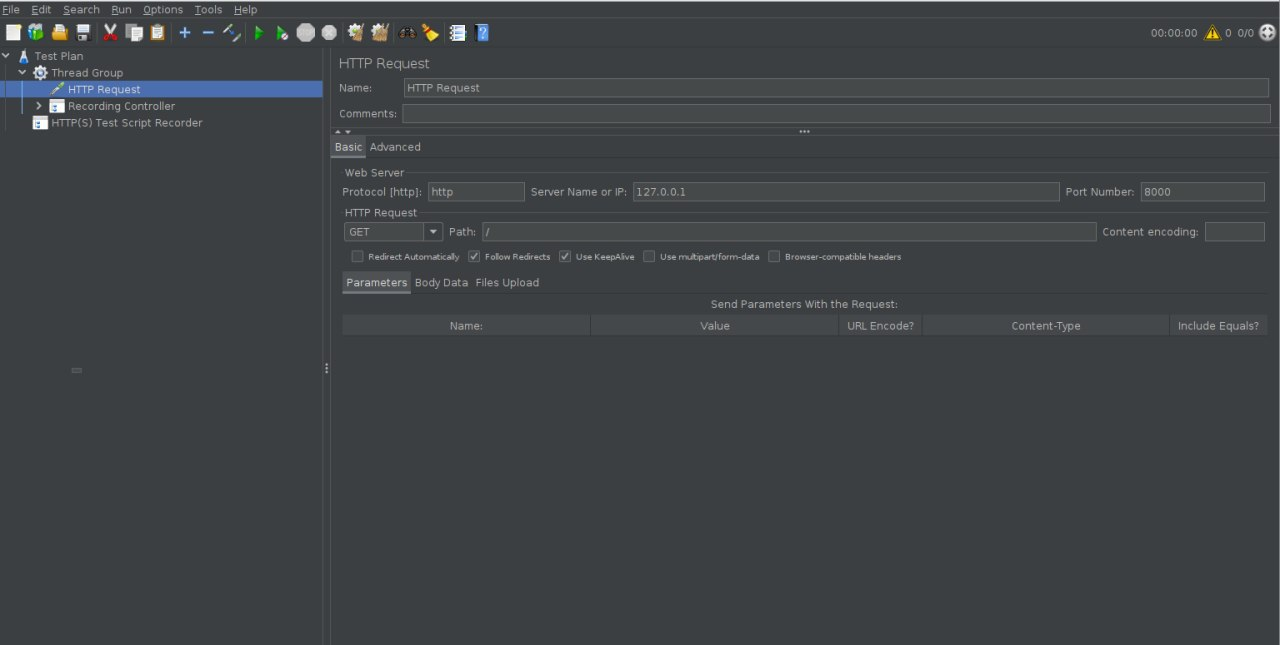


Рисунок 3 – Элемент HTTP Request.

Для записи тестов через проксирование требуется создать Элементы Recording Controller и HTTP(s)Test Script Recorder. Элемент HTTP(s)Test Script Recorder нужен только, чтобы записать запросы. В качестве номера порта прокси-сервера был установлен порт 8888 (рисунок 4).

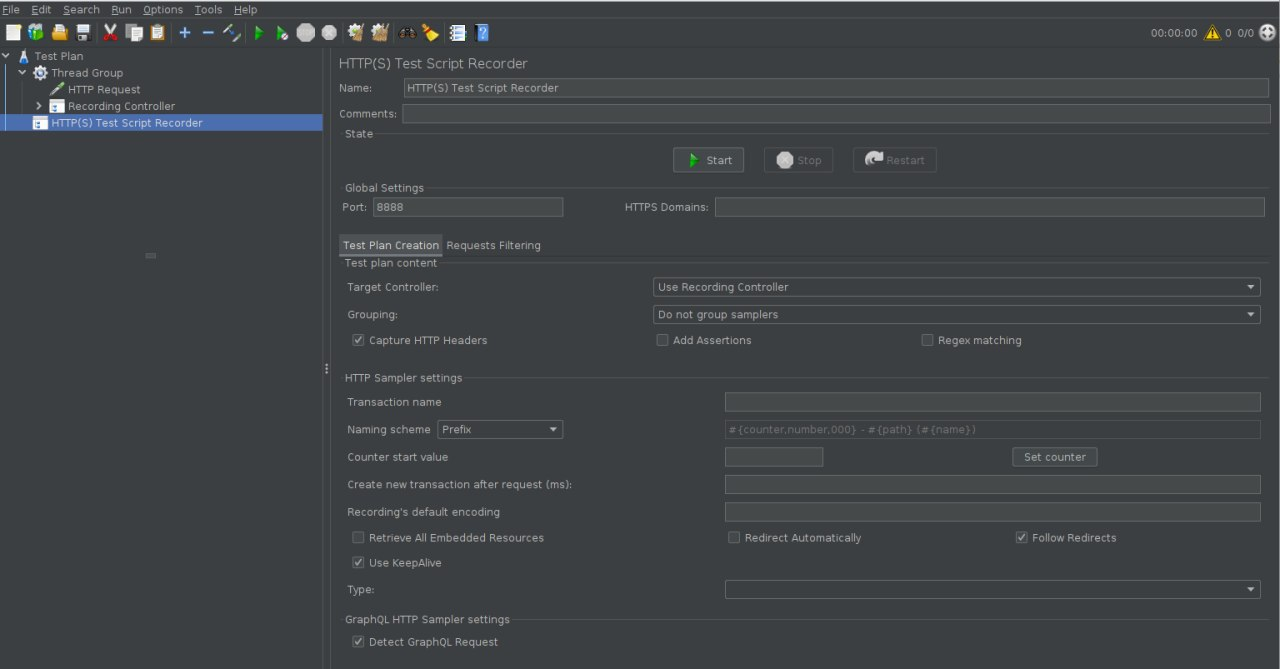


Рисунок 4 – Элемент HTTP(s)Test Script Recorder.

Для корректной работы тестов необходимо изменить настройки браузера. В качестве браузера был выбран Mozilla Firefox. В настройках браузера требуется указать адрес прокси и порт (рисунок 5). Можно убедиться, что браузер входит в интернет именно через нее, для этого перейти на любой сайт в интернете, при этом страница не должна загрузиться.

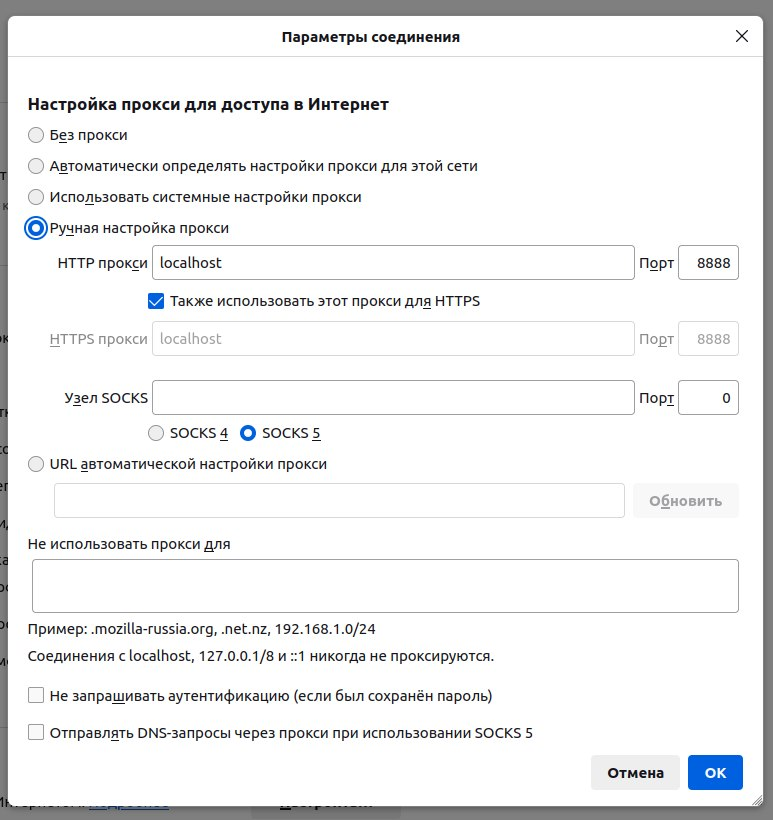


Рисунок 5 – Настройки прокси.

После всех настроек можно приступать к записи тестов. Была произведена авторизация на сайте и выполнены некоторые действия. Все они были записаны в элементе Recording Controller (рисунок 6).

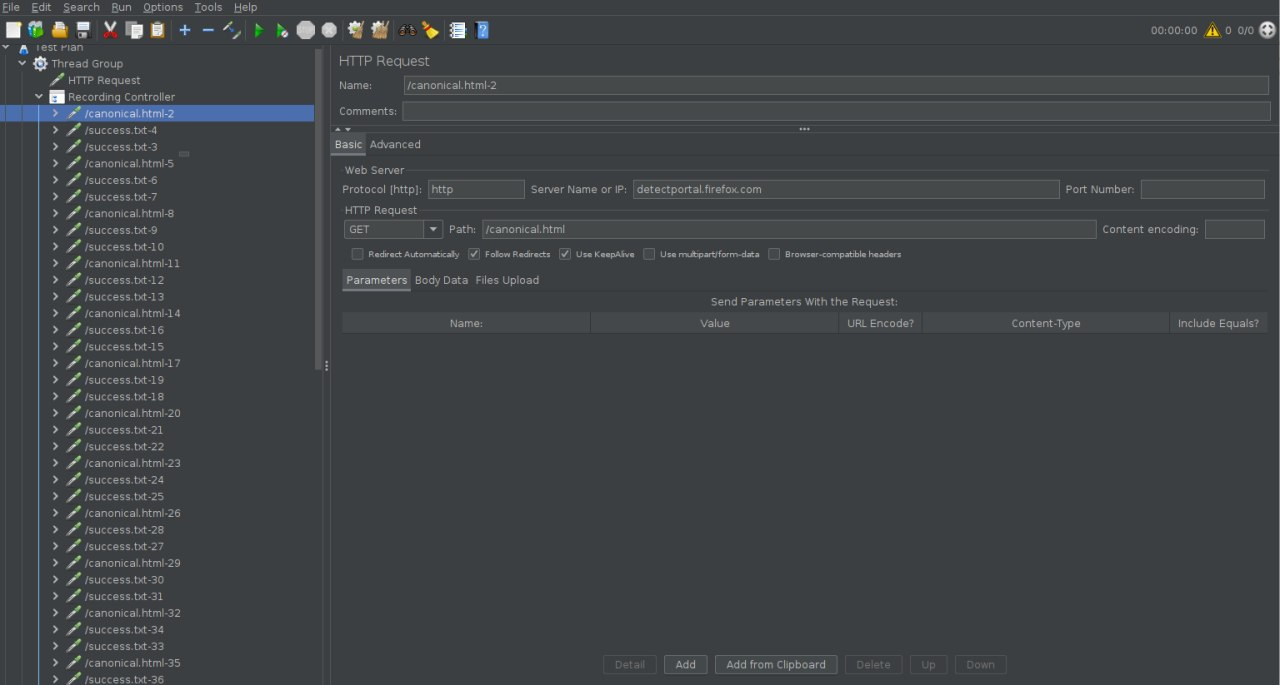


Рисунок 6 – Элемент Recording Controller с записанными действиями на сайте.

Для наблюдения результатов теста, а также для мониторинга хода выполнения данных тестов в ветку Thread Group было добавлено несколько элементов мониторинга: View Results Tree, View Results in Table, Graph Results, Summary Report (рисунок 7).

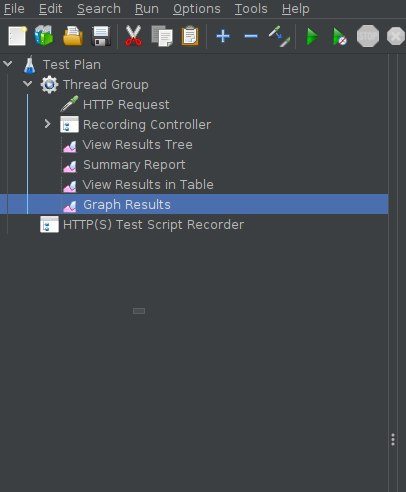


Рисунок 7 – Элементы мониторинга.

Загрузка списка пользователей из файла для имитации того, что некоторое количество пользователей логинится на ресурс, была реализована с помощью добавления элемента CSV Data Set Config. Список пользователей с паролями хранится в файле с расширением csv (рисунок 8).

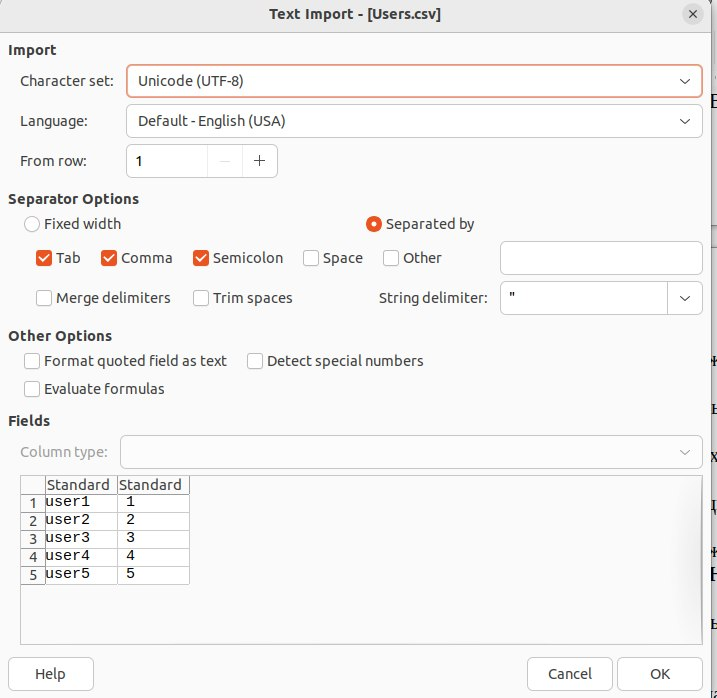


Рисунок 8 – Файл со списком пользователей.

В результате нагрузочного тестирования было выявлено, что сайт требует доработки, так как не обладает должной надежностью и отказоустойчивостью, что видно в элементе View Results Tree (рисунок 9).

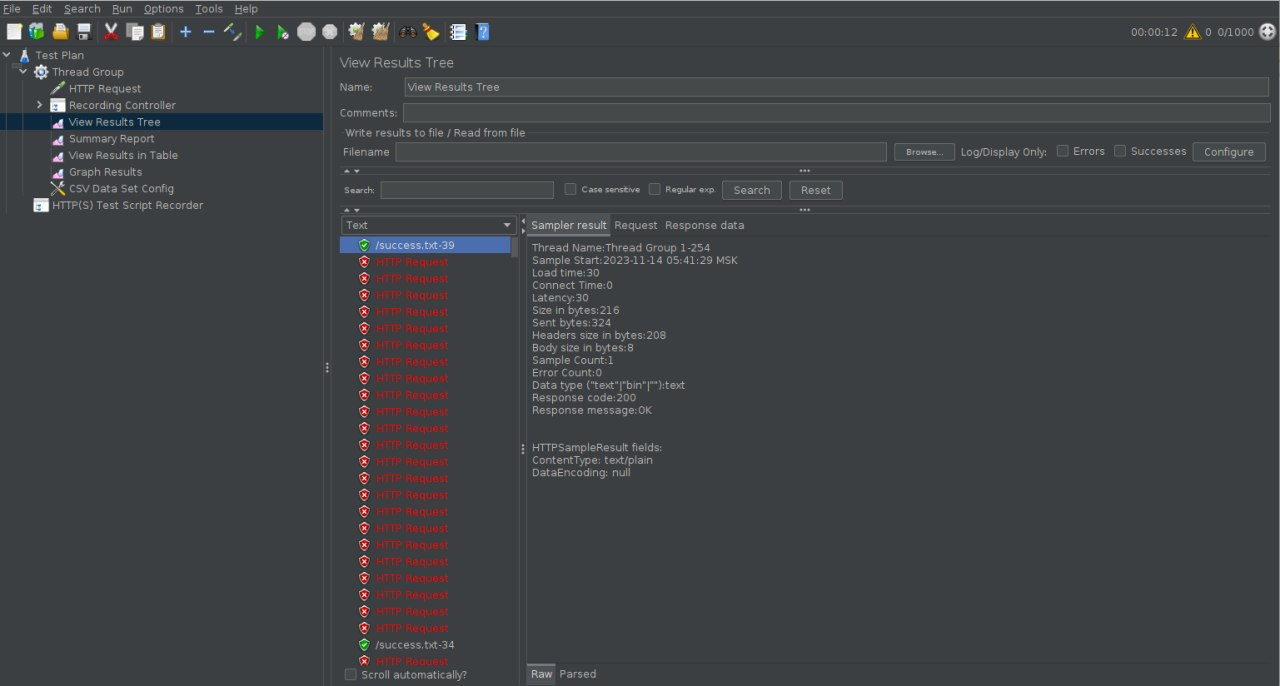


Рисунок 9 – Элемент View Results Tree.

Помимо данных, отображенных в элементе View Results Tree, есть возможность отследить результаты тестирования и через другие элементы: Summary Report (рисунок 10), View Results in Table (рисунок 11) и Graph Results (рисунок 12).

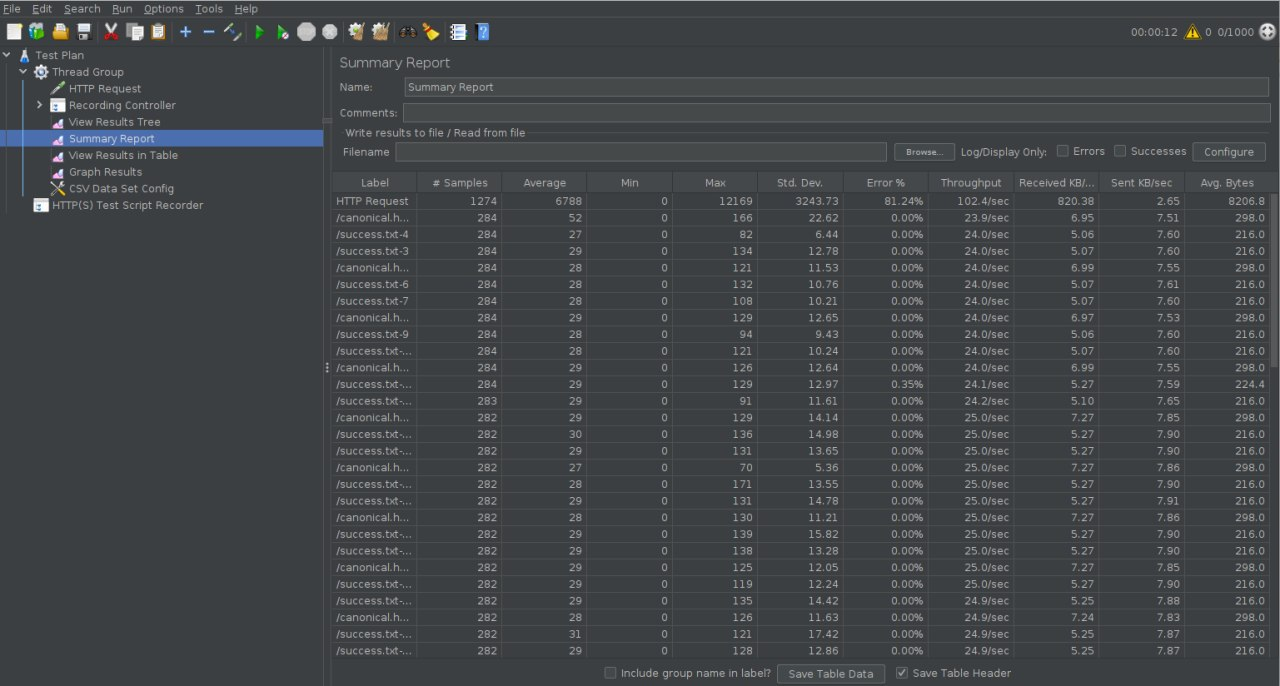


Рисунок 10 – Элемент Summary Report.

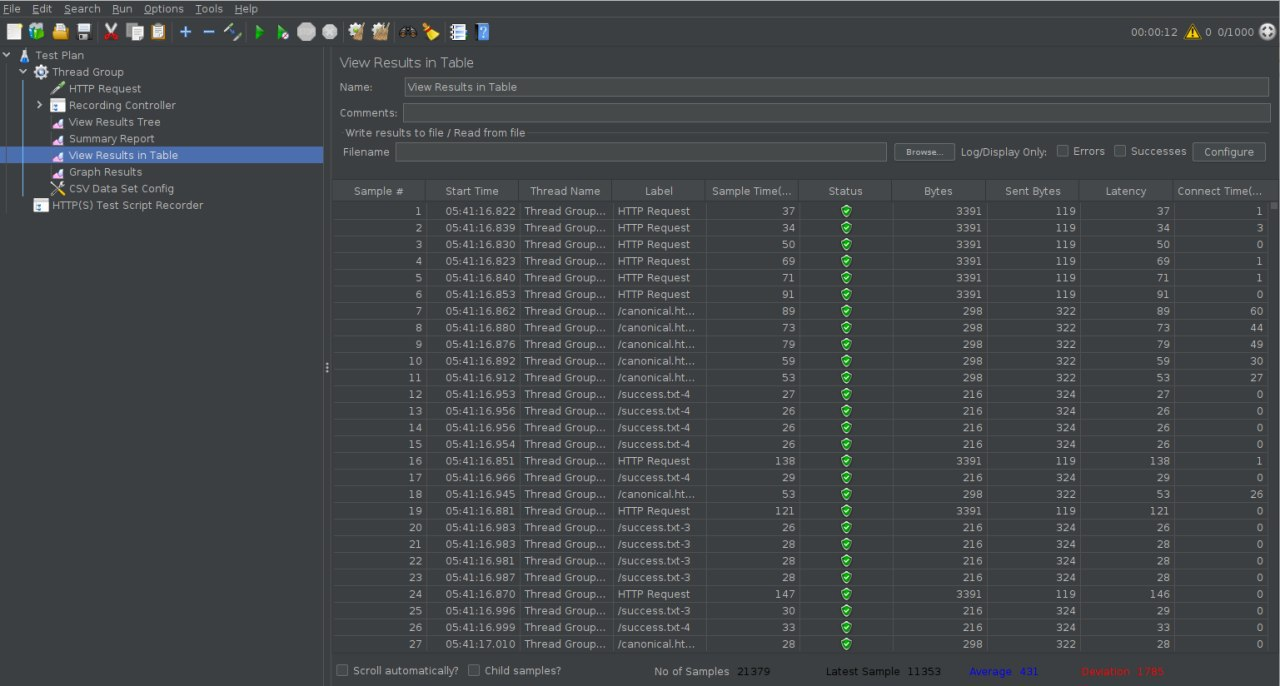


Рисунок 11 – Элемент View Results in Table.

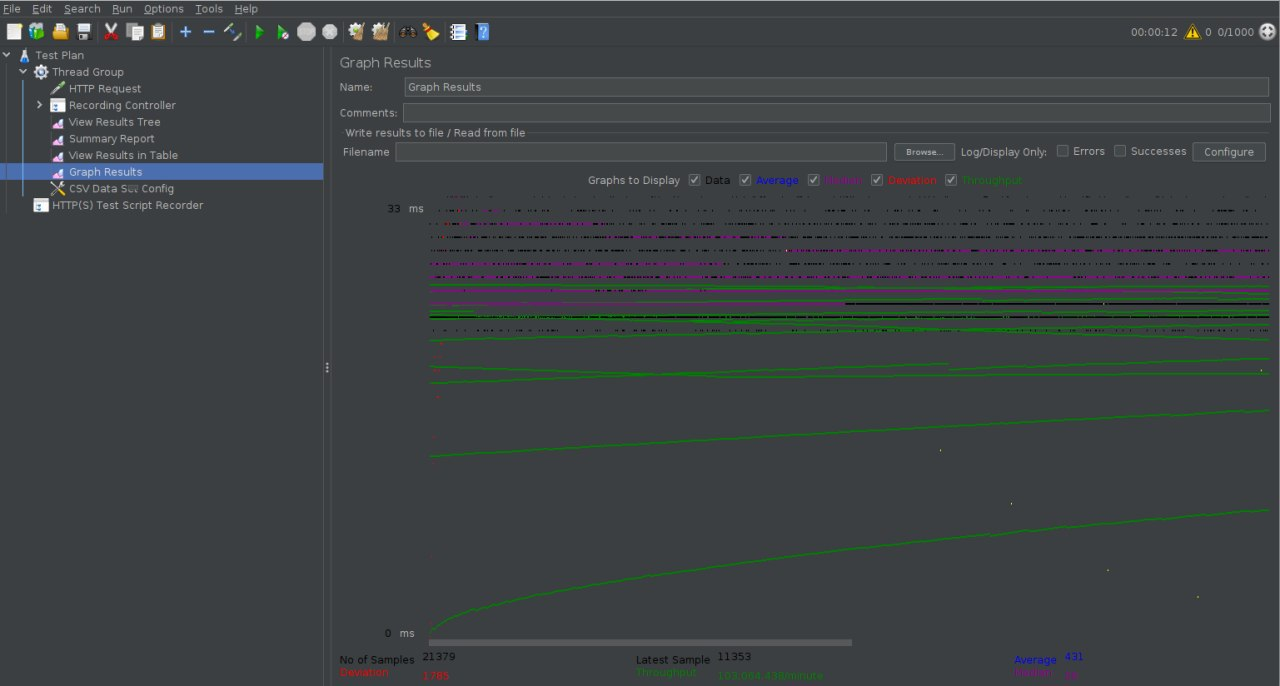


Рисунок 12 – Элемент Graph Results.

В результате выполнения данной работы было проведено нагрузочное тестирование сайта собственного написания на предмет его надежности и отказоустойчивости с помощью такого средства, как Apache JMeter. По итогам тестирования можно сделать вывод о том, что надежность сайта нуждается в доработке, так как сайт не способен обрабатывать большое количество запросов от большого числа пользователей одновременно, что является недопустимым для такого продукта, как система массового обслуживания.