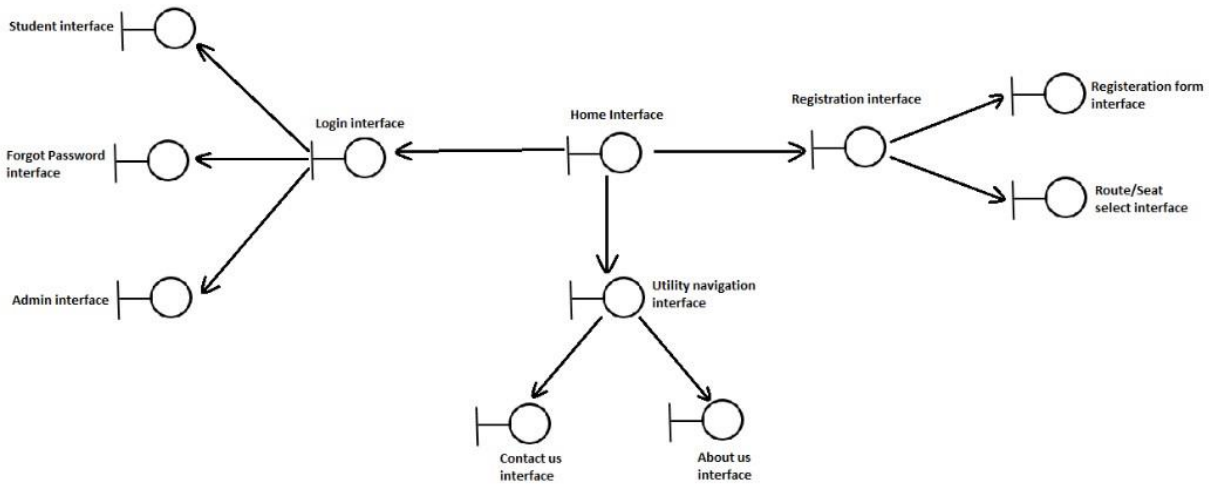
Contents

TESTING REPORT

PROJECT OVERVIEW:

Our website is made to facilitate NUST students in registering for NUST transport system. Students can register online, request for route availability, and after registration, have a portal where they can log in to view their route, van and driver's detail. This website also benefits Admin of the system to keep record of routes, vans, drivers, students and feedbacks given by students.

WEBPAGES:

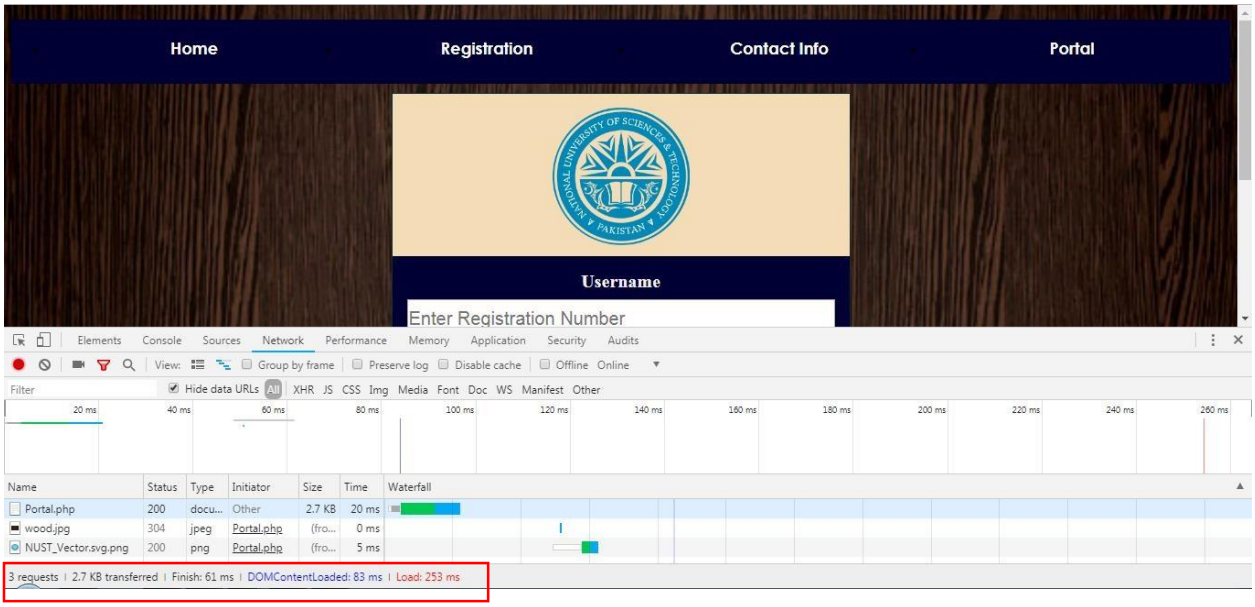


The webpages of main importance upon which the performance and efficiency of the site depends are: Login, Registration, Route request, Student Portal, Feedback, Admin Portal (add, delete, view) pages.

1- NETWORK LOAD TESTNG

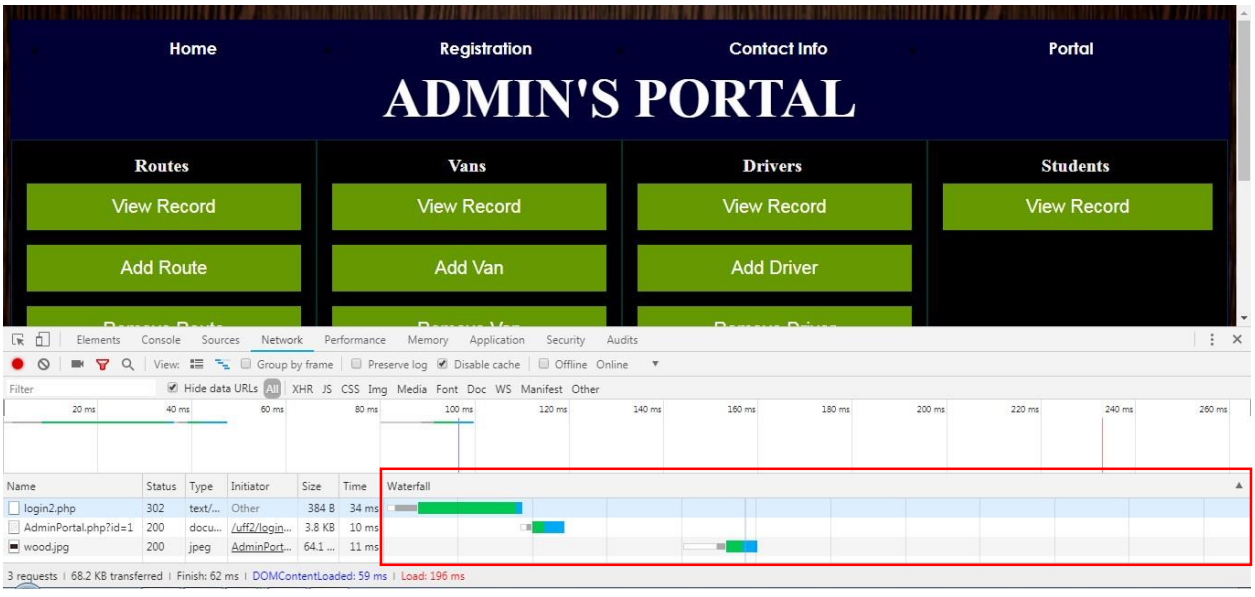
TOOL: CHROME DEVELOPER'S TOOLS – NETWORK

The Network tab of Dev tools is useful in analyzing the network load of individual webpage. It shows the total requests generated by the page, time taken by each request. Below we are testing the login page Portal.php:



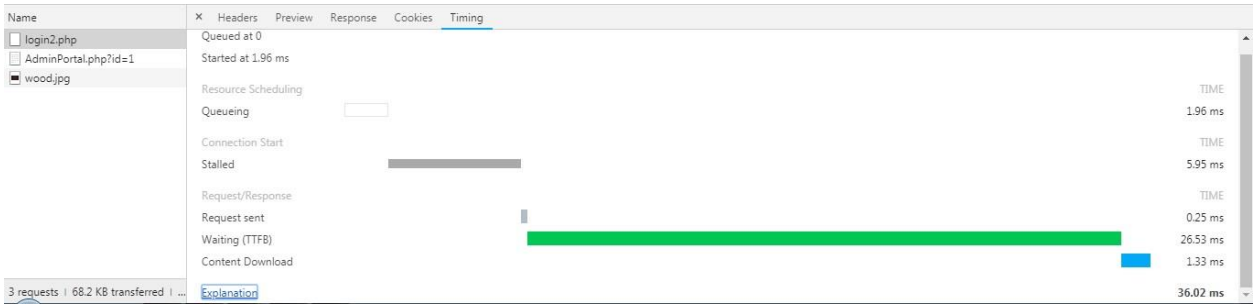
At the bottom of the Network tab is a summary of the page load process. This summary includes the total number of requests i.e. 3 requests, the total size of the entire page including its resources (2.7 KB), and a breakdown of the time spent rendering the page in the browser. DOMContentLoaded is time for page to appear loaded as perceived by user, and it is 83ms. The whole Load time is 253ms; it is more as it also includes the time taken by the background JavaScript actions.

Now we enter credentials and log in as admin. We enter page AdminPortal.php. The network load is as below:



The Network tab shows that Load time is 196ms. The waterfall region is used to view the timing of requests in relation to one another. We can see that much time

was taken by login2.php request, which is the script that validates the login information entered by user. Clicking on the request shows its details as below:



The request was stalled for 5.95ms. The reason for stall is from browser side; maybe there were higher priority requests before, etc. Most of time, 26.53ms is gone in waiting state by request. This is the state where browser is waiting for response. Hence 26.53ms were taken by the server to validate login information and return response to browser. Hence this time is due to the server side, and not due to website.

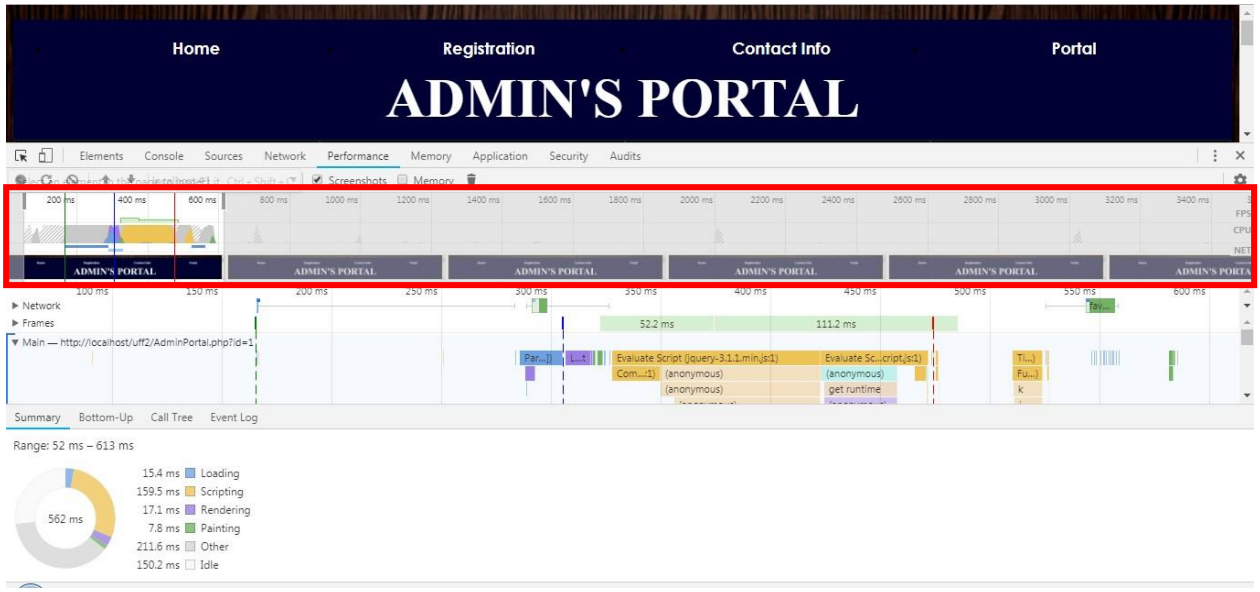
RESULT:

There have not been observed any long loading time. Any waiting s due to the server side performance.

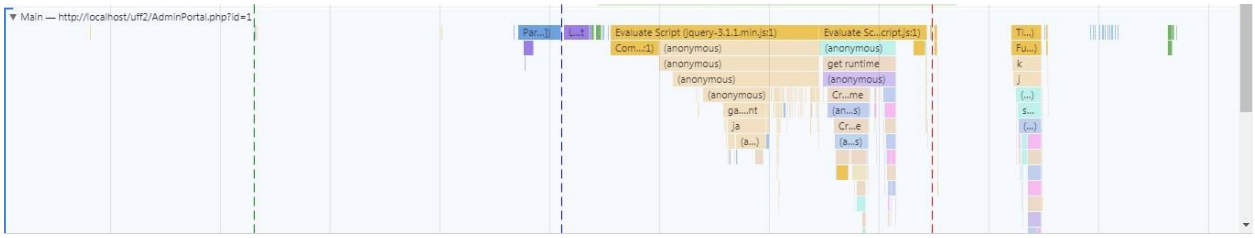
2- PERFORMANCE TESTING:

TOOL: CHROME DEVELOPER'S TOOLS – PERFORMANCE

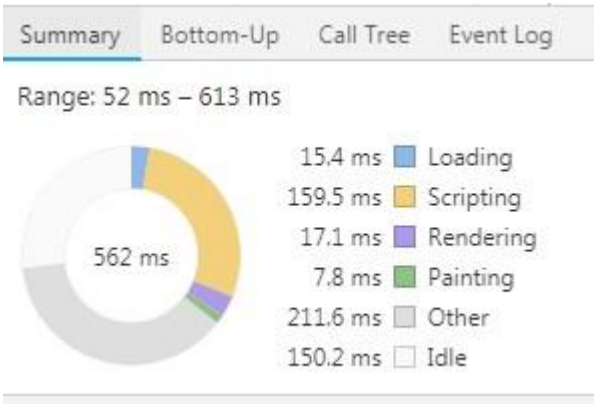
The Performance tab is used to analyze the runtime performance. Runtime performance is how your page performs when it is running, as opposed to loading. This tool animates the page in frames per second FPS. We have analyzed AdminPortal.php. The highlighted region is the animation in FPS recorded by the tool.

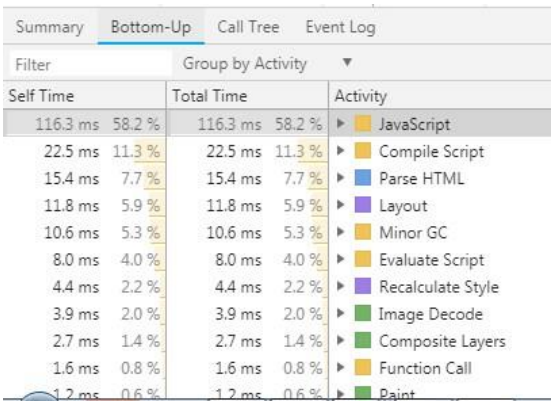


In the Main section, DevTools shows you a flame chart of activity on the main thread, over time. The x-axis represents the recording, over time. Each bar represents an event. A wider bar means that event took longer. The y-axis represents the call stack. Hence, the yellow bar i.e. Evaluate script bar activity took the longest.



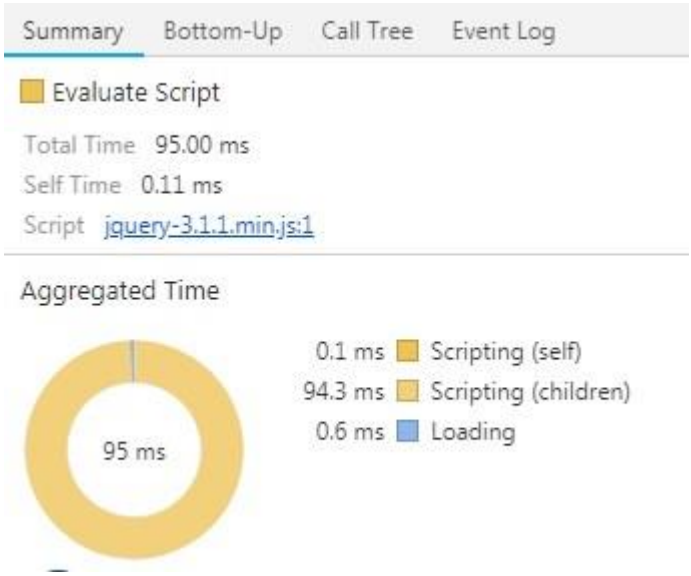
We can see the summary of activities in Summary Tab. Also Bottom-up Tab shows activity in order of longest to shortest.





RESULT:

Scripting takes most of the time. It is inevitable. As it is the parent process, necessarily called to run other children functions on the page.



3- VALIDATION TESTING

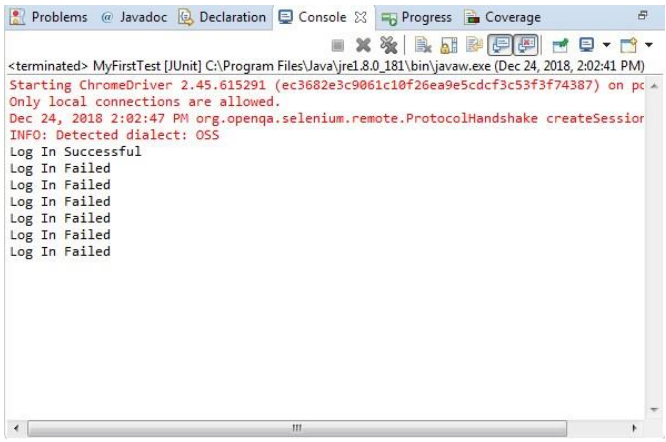
TOOL: SELENIUM WEBDRIVER

Selenium WebDriver is a tool for automated testing of browsers. It facilitates easy identification of elements on webpage, inputting data, and validation. We have used Selenium WebDriver to validate the login page i.e. we provided it with valid inputs and also invalid inputs and then Selenium ran automated tests on the page.

Selenium can also be used to input mass random data, but we used Selenium just for validating login functionality. (The code is included in the submission).

RESULT:

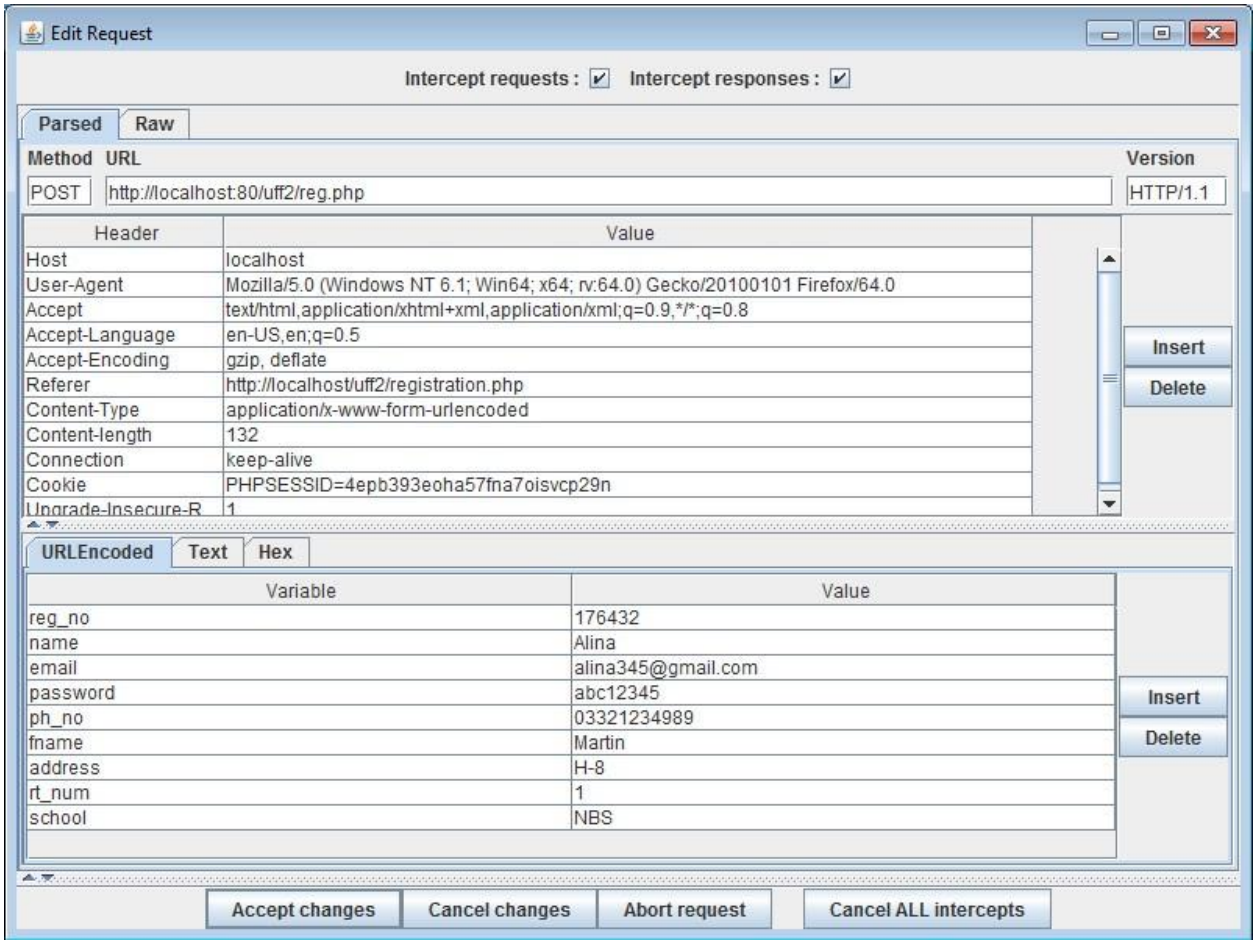
Selenium has validated login functionality of our website against the data inputted by us.



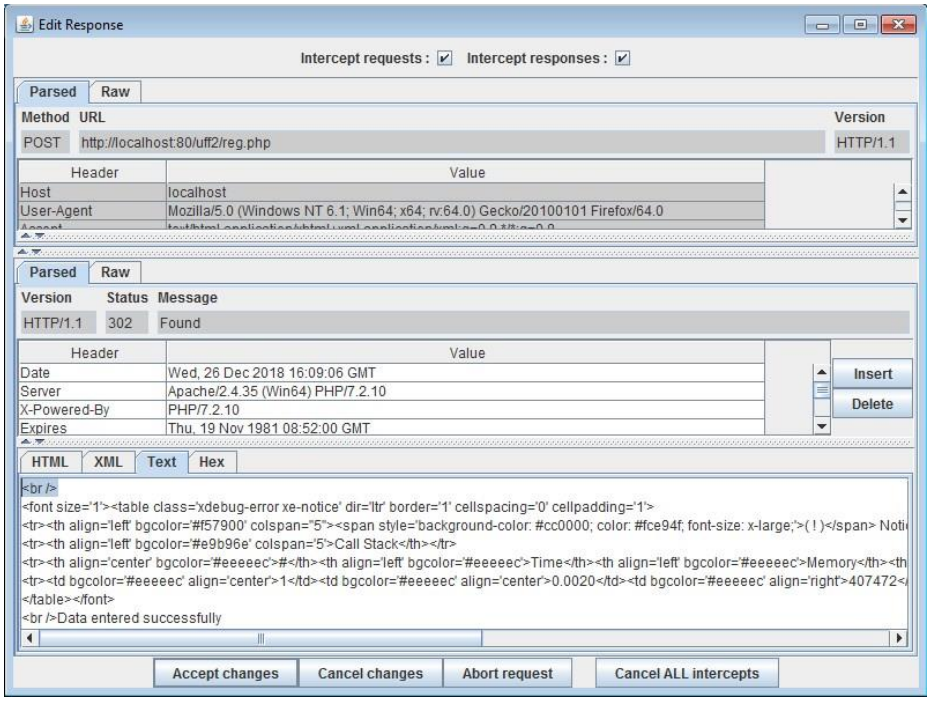
4- FUZZING

TOOL: WEBSCARAB

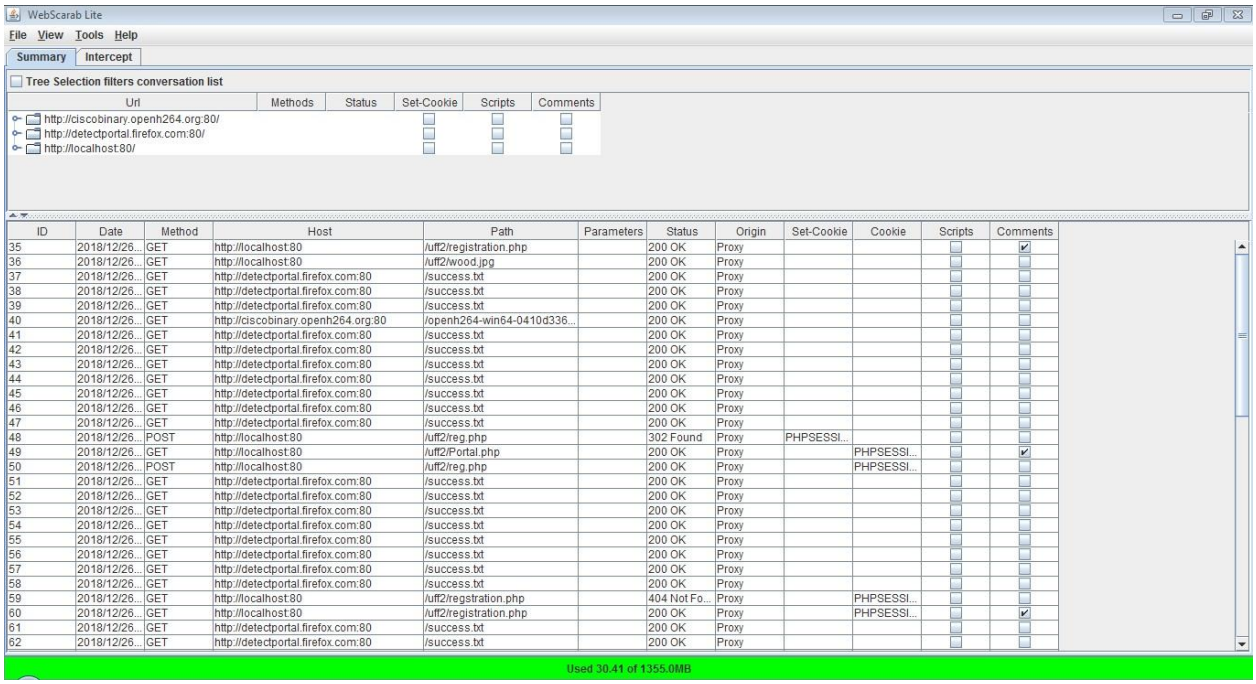
WebScarab has been used for fuzzing. Its parameter fuzzer automatically substitutes parameter values hence incomplete parameter validation can be exposed. We performed fuzzing on registration form. We started with allowing WebScarab to intercept GET and POST requests for the registration form. Then it starts filling out the form. WebScarab also allows us to review and modify request created by the browser before they are received by the server (when we enable this). For example, here we modified the request to contain proper school, Islamabad sector, and phone number as we desired:



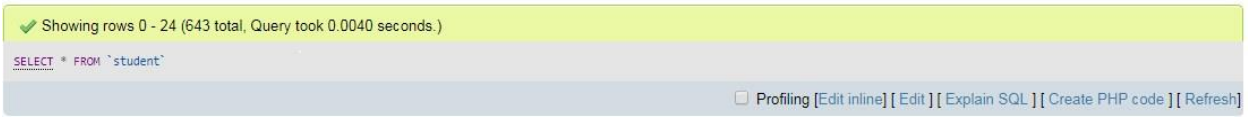
WebScarab also shows response.



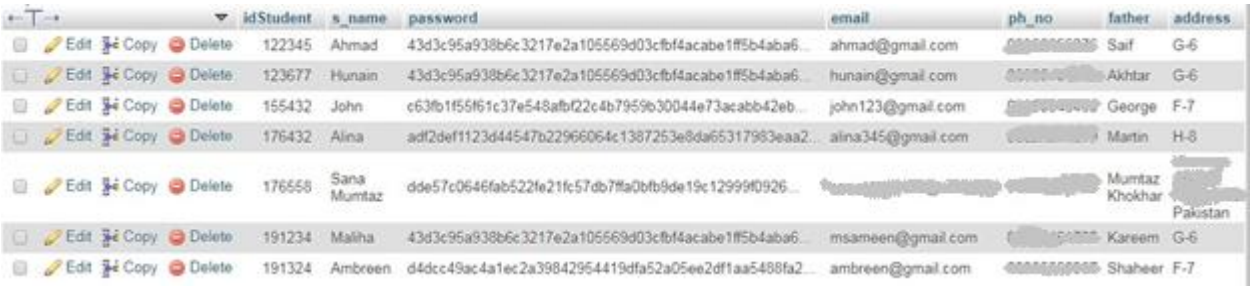
Here is overall input, success, and response messages.



Data was registered almost 635 times. The amount of data registered as shown by database:



And this is how the data looks in database. (Passwords are encrypted).



RESULT:

WebScarab allowed easy fuzzing on registration page.