

## Progress Report

Group 20

By Linda Chang, Yuchen Huang, Yue Guo  
(our fourth member never responds or shows up)

### Introduction:

Our client and his team in Teradata are working on a large, open-source project named Presto. The project uses GitHub and has many repositories. The Teradata team would like project status reports generated of their progress in the Presto project, beyond what GitHub currently offers. For example: Average time span from open to close. Average number of commits and lines of code. Contributor details: individual and organization.

### Target Users:

the users would be a github user who wants to see more stats than what is currently available in github another user may be a power use that wants to add more custom graphs and charts than that is precanned in your app

### Goals:

We will get data of the Presto's project status and reports from GitHub, categorize and filter the information, and display the results clearly on a web interface.

Ideally, we would also like to use data visualization to display the requisite information in widgets, and implement an interface for the user to be able to specify the parameters of the data they want displayed.

### Functionality Request from Client:

1. Pull Request details issued from an organization/user to a specific repo. Includes basic details. Choose open, closed, or both. Specify date range.

We are now able to build a server and pull request details from the API when the user enters the name of specific repo, such as "presto". We are receiving all the data we need to construct the following steps of displaying data. Since the client mentioned what he need was the information, we are now displaying the requested data simply in a table. We also added a login system so every user could use their own personal token and then get access to the API (login is currently in progress, not done yet). The next step for us is to filter the data and delete the unnecessary information as well as display data in a prettier way, and improve the login system more user friendly. Though he considered the data visualization as extra credit, we still want to display the data using d3 (most possibly in a timeline and histogram format)

2. Variant — Average time span from open to close

We also need to fix some problems because now we could only get data with state "open". It seems that we could not send parameters in order to get data of both states of "open" and "closed". We are now able to get data of open states, and we are receiving the starting time but no ending time because the states are all open. If we can figure out how to get the data of closed states, we should be able to get the time span with its starting and ending time.

3. Variant – Average number of commits and lines of code

Since we have already had the data, we just need to implement counters in order to count the lines of commits and lines of code. We are planning to display the data not simply in text but

also in a chart using d3. It would most probably be a timeline or histogram chart corresponding to the user's selection.

#### 4. Contributor details: individual and organization.

This is not requested by the client, but we think it would be really good if we can display the individual and organization profile as well. We have received the url links, so we just need to modify the data and let it display on the page nicely.

#### 5. Client feedback:

We have already visited the client whose office is located in Boston and specify what we had done and planed to do. From now on, since we could almost achieve some functionality, we are planning to contact him more often depending on the progress(Ideally more often than once a week).

#### Technologies & Frameworks:

GitHub API to pull out the necessary data.

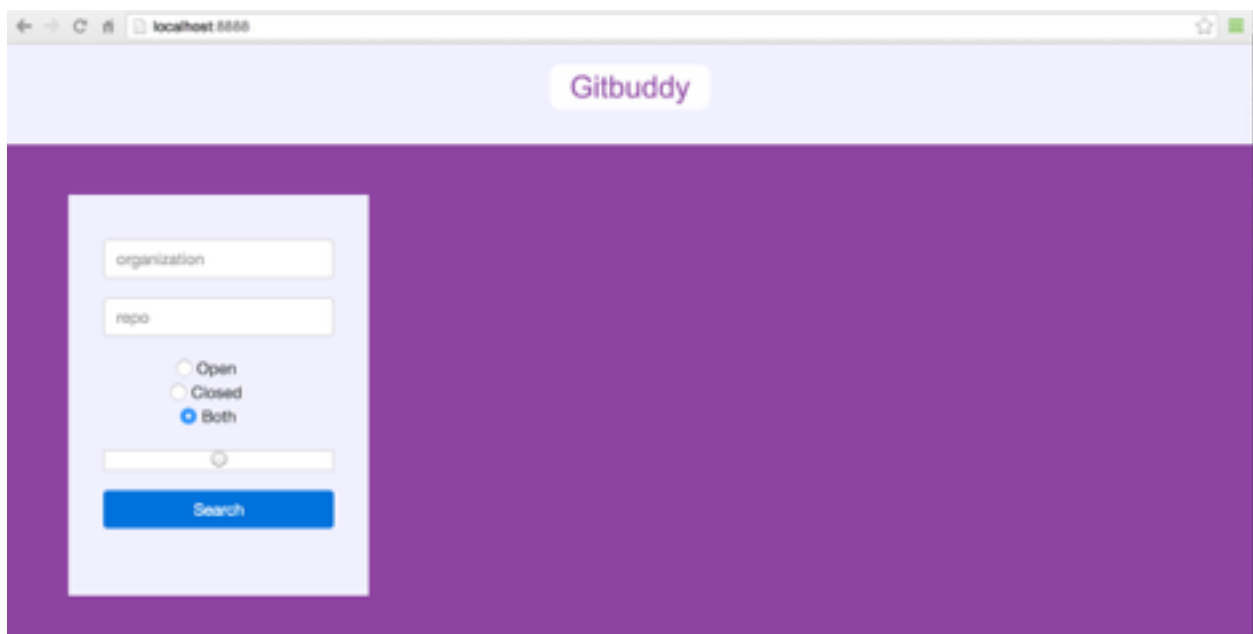
Node.js for the back-end.

D3.js for data visualization.

HTML/CSS/Javascript web front-end.

SQL to connect to the database and access the data with different properties. (Originally we think we need the SQL and database, but now it is not an issue because we are directly pulling data from the Github API and not storing anything)

Enclosed is the screen shots of what we have realized:



localhost 8888

Gitbuddy

Teradata

presto

☐ Open
 ☐ Closed
 ☒ Both

Search

state	closed_at	comments_url	commits_url	created_at	diff_url	html_url	id	issue_url
open	null	https://api.gi	https://api.gi	2016-04-21T12:20:36Z	https://github	https://github	67344392	https://api
open	null	https://api.gi	https://api.gi	2016-04-20T12:15:10Z	https://github	https://github	67175996	https://api
open	null	https://api.gi	https://api.gi	2016-04-18T15:41:48Z	https://github	https://github	66875396	https://api
				none ne				