

Assignment #4

Assignment Due Date: 4/14/19 by 11:59pm

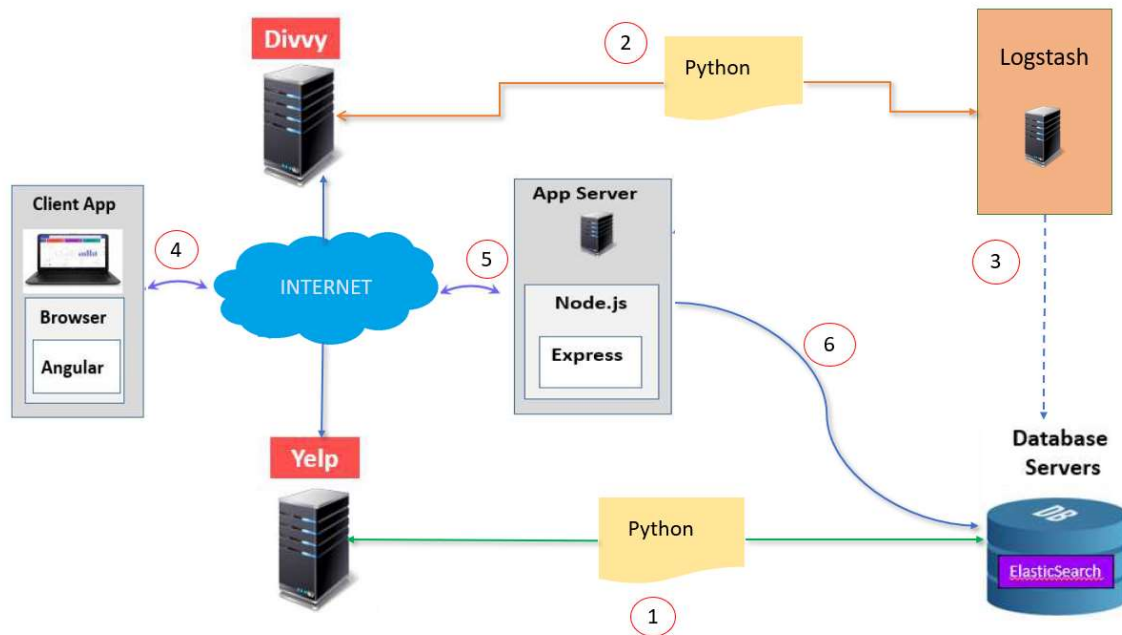
Assignment Deliverables:

You are required to submit a SINGLE WinZip file that has the following deliverables :

1. Documentation for the Angular Components that you created to draw the Line chart
2. Source Code
3. Output report that has ALL captured screen-shots of your assignment run saved in OUTPUT.pdf
4. Video recording of 10 minutes as a demo for the run of your assignment using <https://screencast-o-matic.com/>

Post your assignment as a SINGLE WIN-ZIP on Blackboard.




Architecture/High-level Design:



Requirements:

Refine your Assignment #3 implementation of ChicagoSocialHub to achieve the following :

- 1) Use logstash to log Divvy the periodic divvy status on ElasticSearch server. Your log MUST accumulate data of at least 7 days.
- 2) Use and pdate the provided logstash.conf filter to clean entries that have white space after/before data values; for example a white space before/after Chicago in city name
- 3) Use the provided Python script to collect data from Divvy and send them to Logstash
- 4) The periodic Divvy heart beat will be stored on PostgreSQL server
- 5) A Line Chart to plot the number of available docks of the selected divvy station.
- 6) The user shall be provided with a button next to every nearby Divvy station listed to view/select options for the Line Chart. The following is an example of a valid listing/display:

ID	stationName	availableBikes	availableDocks	is_renting	lastCommunicationTime	latitude	longitude	status	totalDocks	
197	Michigan Ave & Madison St	6	13	true	2019-03-31, 9:30:11 am	41.882134	-87.625125	IN_SERVICE	19	Real-Time Chart 
43	Michigan Ave & Washington St	2	41	true	2019-03-31, 9:32:12 am	41.8838927658	-87.6246491409	IN_SERVICE	43	Real-Time Chart 
39	Wabash Ave & Adams St	5	18	true	2019-03-31, 9:30:43 am	41.87947235235	-87.6256886059	IN_SERVICE	23	Real-Time Chart 

- 7) After the User clicks the Real-Time Chart button, a new page is displayed where the user is provided with a dropdown buttons/list labeled **Time Range** . The user shall be provided with the **Drop-Down list** with label "**Time Range**" in order to choose 1 hour, 24 hours, 7 days in Green color.
- 8) The Line Chart must be updated in real-time without a MANUAL refresh.
- 9) When you plot divvy data on your Line Chart,
 - i. the X-axis shall display time in granularity of 2 minutes (30 label/interval total displayed)
 - ii. the X-axis shall display time in granularity of Hours (24 label/intervals total displayed)
 - iii. the X-axis shall display time in granularity of days (7 label/intervals total displayed)
- 10) Right next to "**Time Range**", the user shall be provided with **TWO Check-BOX Buttons** to draw/overlay on the **same chart** the 30 (past 1 hour divvy data) simple moving average in Blue color and 720 (past 24 hours divvy data) simple moving average in Red color. So, on the same Chart, the user will see real-time data (Green line) , 1 hours SMA (Blue line) , 24 hours SMA (Red line)
- 11) The data collected from divvy stations using the provided Python script must be stored in an index you create on Elasticsearch server with the name divvy_stations_logs
- 12) Every page must have a **Back** button display on the Far-Right
- 13) Every page must have a **Home** button display on the Far-Left; this button always will take you to ChicagoSocailHub App homepage
- 14) All buttons shall be displayed on one line below the Chicago Social Hub Logo
- 15) Buttons and icons must be displayed evenly on your UI
- 16) Create the UML sequence diagram to plot the line chart
- 17) Create the UML design class diagram for the application