### **Task #174062, Programming & Computer Science, AJAX | JavaScript | HTML | PHP**

Project consist of **3** files: index.html, realtime-chart.js and db.sql. Project demonstrait streaming graphing of bar chart.

**Database**

**For this project were used relational database MySQL.**  
  
**To create a table use script below:**

CREATE TABLE transactions(

t\_id INT NOT NULL AUTO\_INCREMENT,

t\_time INT NOT NULL,

t\_data INT NOT NULL,

PRIMARY KEY (t\_id)

) engine MyISAM DEFAULT CHARSET=utf8;

**To fill a table with date you should use stored procedure wrote below:**

DELIMITER $$

CREATE PROCEDURE prepare\_data()

BEGIN

DECLARE i INT DEFAULT 100;

WHILE i < 1100 DO

INSERT INTO transactions (t\_time, t\_data)

VALUES (UNIX\_TIMESTAMP(NOW()) + i \* 10000, (FLOOR( i + RAND( ) \* 5 )));

SET i = i + 10;

END WHILE;

END$$

DELIMITER ;

Short description of procedure: Default value for counter **I** set to 100. We will increment this value by 10 so that at the end we will got 100 records in DB. In while loop we perform insertions to table. **t\_time** value is a unix timestamp plus a delta (delta needed to increase intervals between previous and next time values). **t\_data** will be an integer random values in range from 100 to 1000.

**To perform stored procedure please follow instruction below:**

CALL prepare\_data();

**Server side**

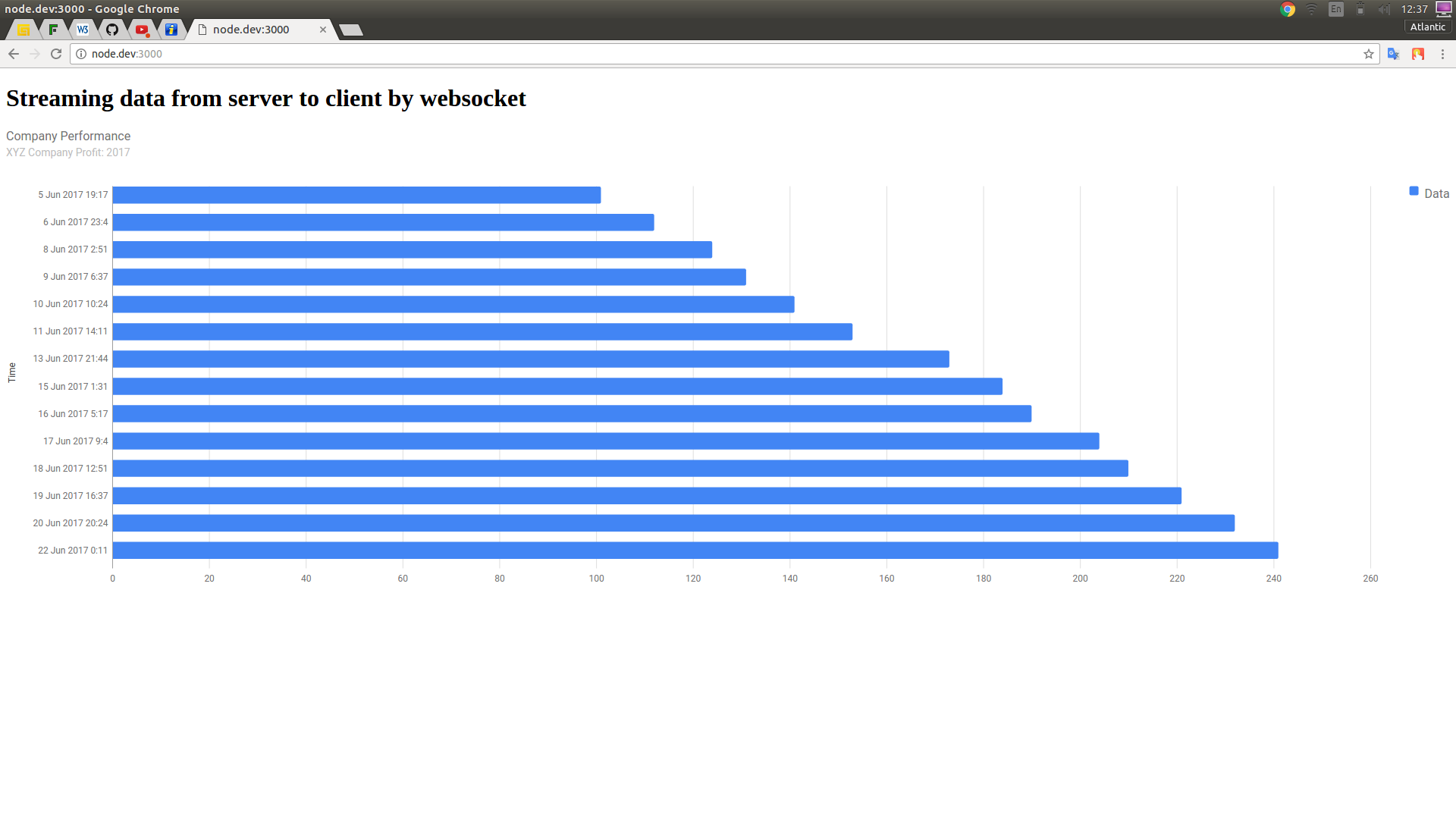
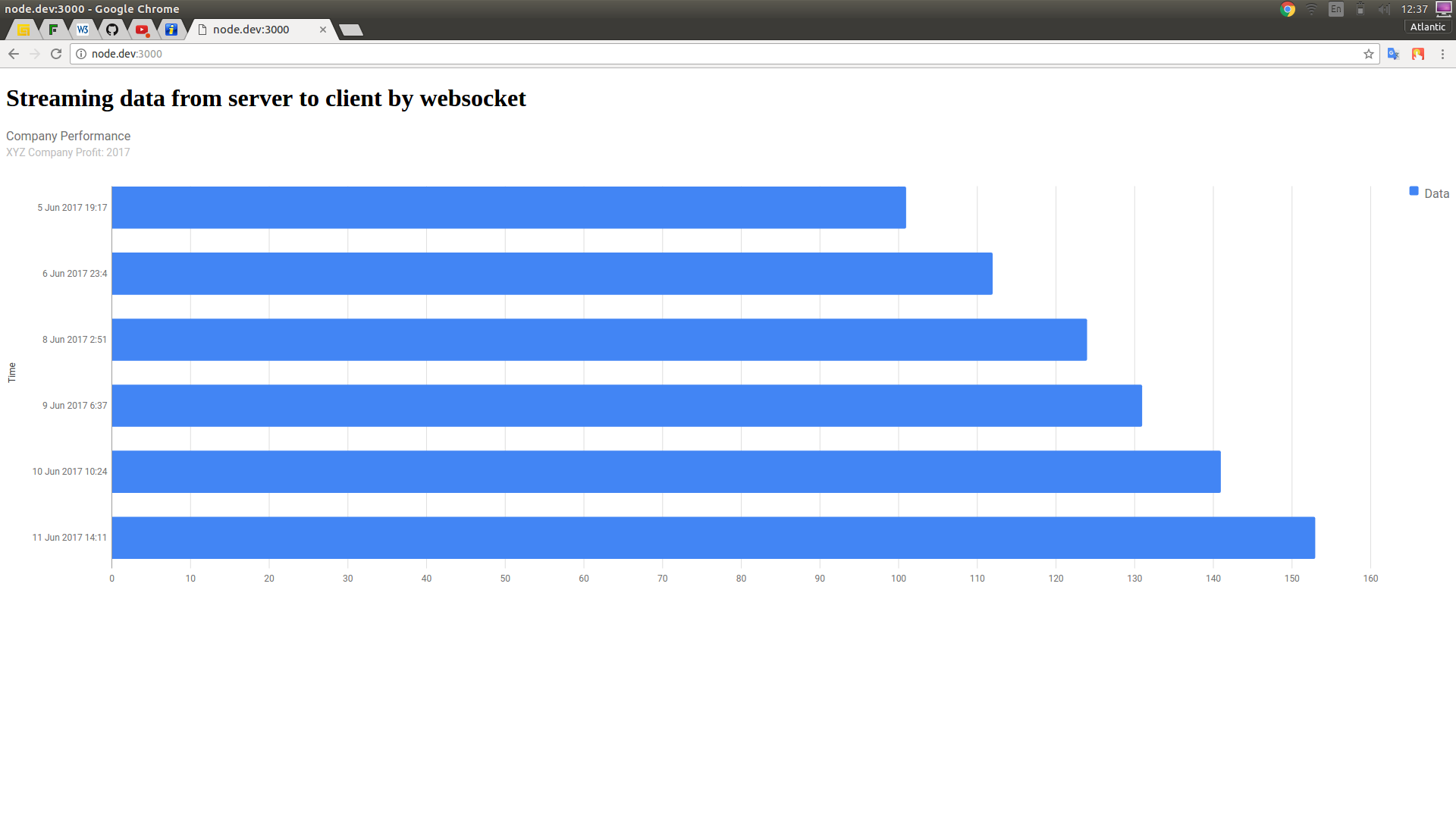
For server side was used node.js of version 6.9.2 with socket.io and mysql third-party modules. Also were used http and fs build-in modules.

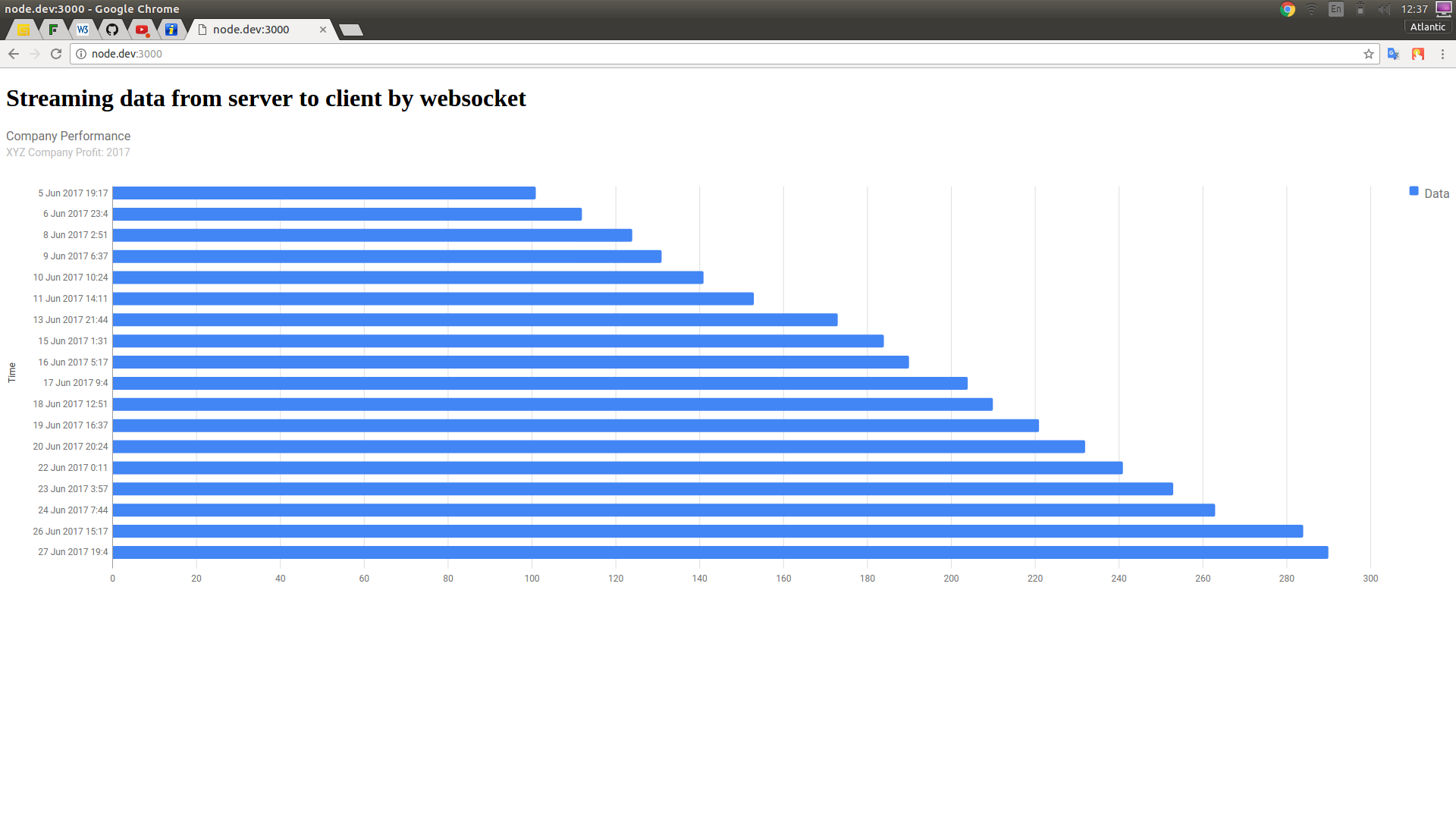
Before run project in browser you should create DB and edit connection settings to it in realtime-chart.js file.

Program imitate realtime incoming data by using **setInterval()** function during querying DB. Than data reformated to JSON and than be pushed to client. Comunication between server and client built on socket API.   
  
 **Client side**

Client side utilizes Google Chart API. Also there is snippet utilizes socket. On UI JSON has been parsed and rendered.

Screen shots of a program:





**Instructions to run node.js application (Ubuntu-based):**

- Go to project directory;

- Exec command “sudo apt-get install nodejs”;

- Exec command “npm install socket.io”;

- Exec command “npm install node-mysql”;

- At this point you also should have MySQL server on local machine;

- Add info to connect to MySQL DB server in realtime-chart.js file;

- Perform command “node realtime-chart.js” in console in project directory;

- Change line “[http://node.dev:3000](http://node.dev:3000/)” to “[http://localhost:3000](http://localhost:3000/)” in index.html;

- Go to browser and visit url “[http://localhost:3000](http://localhost:3000/)”;

- You should see a chart.