Team Name: vincem
Team Members: Vincent Mattingly (10991225)
Team Id: c13e75ad-357f-4d09-8d91-d9f0a7b9a8c5

Individual Contributions: I did everything so here's a summary of what's going on:

I'm using NodeJS + MySQL. NodeJS has an http library built in but I had to install the mysql driver using npm. There are three main files in the project: projectschema.sql which wipes the database and resets the schema when redirected into mysql, Project.js which handles all the http POST/GET api's, and testsnippets.sh which is used to automate testing.

Most of the interesting stuff happens in Project.js. At the very bottom I call http.createServer(httpListener).listen(9998) which binds the function httpListener to port 9998. httpListener uses a regular expression to decompose the URI into smaller portions: "/api/<function>[/param]". In then waits until the full command has been read. Once the entire command is known all relevant information is passed to the handleRequest function.

handleRequest determines if the desired command is implemented on the server, if so it parses the json request and URI parameter if relevant and passes it to whichever command was requested. These commands for the most part follow two patterns: write the result of a query to the response stream and close it, or write whether or not the query was successful to the response stream and close it.

Running the query is largely handled by the connectAndQuery function, which connects to a mySQL server database called "project". The query is preprocessed to remove whitespace and comments which makes the error messages easier to read. If everything goes correctly, it calls a callback function, typically either writeObjectToStreamAndClose or writeSuccessToStreamAndClose, with a minor amount of preprocessing to ensure the entire list is not written if multiple answers are returned.