

## Project proposal

**Team Member :** Myself

**Topic :** Emulating Firebase

I plan to work with traffic collisions data in Los Angeles in the last 10 years. In my project, users can learn about traffic accidents for e.g., the number of accidents that took place in different areas of Los Angeles, time when the accidents occurred, ages, genders and descents of the victims, can find the number of accidents in different years, know about the accidents which took place in the adjoining zip codes of his area etc.

**Planned implementation :** First I plan to collect a big dataset containing traffic data about different areas in Los Angeles.

After collecting the data, I will store the data in JSON format to a MongoDB database with a schema name and an index. This database will now be used for the next tasks.

Next I plan to create a database server(prototype system) that will support all RESTful services and emulate the functioning of Firebase database. It will support all CURL commands(PUT, GET, POST, PATCH, DELETE, filtering functions like orderBy, limitToFirst, limitToLast, equalTo, startAt, endAt) executed from the terminal. Users can query/update/fetch data from the command line using the CURL commands as mentioned above. The output will be displayed on command prompt for each request.

Now I plan to use Flask.js and Websocket code to make the database prototype server hear the requests PUT, GET, POST, PATCH, DELETE, filtering functions like orderBy, limitToFirst, limitToLast, equalTo, startAt, endAt etc. sent from the client at a given port. The server will hear the request, process it, execute the commands and show the results back to the client. The results will be displayed to the command prompt screen.

### Planned Implementation and Timeline :

Planned Implementation	Expected Timeline
Collect a big dataset containing traffic collisions data about different areas in Los Angeles	February 25 (Could be increased based on availability)
Store the data with a proper index to the MongoDB database in JSON format.	March 15
Execute all CURL commands like GET, PUT, POST, PATCH, DELETE, limitToLast, filtering commands etc. from the command prompt and see the results on terminal.	April 10
Create a database server(prototype system) that will support all RESTful services and emulate the functioning of Firebase database. Along with the CURL commands(PUT, GET, POST,	April 20

PATCH, DELETE, filtering functions like orderBy, limitToFirst, limitToLast, equalTo, startAt, endAt) executed from the terminal, I plan to use Flask.js and WebSocket to hear the client requests at different ports, process them and show the results back to the client in JSON format (like Firebase).	
Final assimilation: Show all necessary CURL commands execution from command prompt interface, hear requests from client using Flask and Websocket, query the MongoDB database accordingly and display the results on terminal.	April 24

To be noted : the timelines are provided based on my initial planning while I submit my project proposal. Based on the availability of the data or the feasibility of the tools the timelines for different tasks could be extended little bit. Anyhow I will try to stick to the schedule to my best.