**Implementation**

The whole implementation has three parts- the database for storing data, the web api for data communication and the UI for presentation.

For the database we used MySQL RDBMS (Relational Database Management System). MySQL was chosen for its simplicity and ease of use. It is easy to install, use, as well as scalable and manageable.

For serving the client we made a web API that receives client requests and serves JSON data (search results, citaion data). This web API is written in Python. The reason we chose Python is that during the initial phase of our project we had to try out many prototypes before actually selecting one for the project. Python is very suitable for designing prototypes as a minimal code can produce a working implementation. Not only is it concise its also easy for others to read.

We also used the Flask framework for making the web API. Flask is a web microframework for Python. We considered Django but realized it would be too heavy for our project. Flask is a lot more lightweight framework and is easier to use.

Our front end consists the usual HTML, CSS and JavaScript. For building the citation graph we used cytoscape.js library. There were quite a handful of graph libraries to chose from but this library showed well backward compatibility with older browsers and hence we selected it.

Design changes:

Initially we thought of scraping google scholar for a paper’s metadata and citation data. But using this approach would mean we had to make huge number of requests to google scholar. As google scholar disallows bots we would eventually end up being blocked. So we changed our approach to extracting data by using both google scholar and parsing pdf of the papers.

UI- Aquib, Eusha

API- Rafed, Moumita

Graph- Aquib, Moumita

**Source code management**

To work collaboratively we used Git to manage our project. Our project is hosted at github and can be found at github.com/rafed123/showme. It has been configured to use Travis CI.