GENETIC VARIANT VIEWER

The project I’m proposing is a genetic variant viewer that allows the user to explore known human genetic variants along with the associated phenotypes using data from clinvar and tagging with it descriptions of the phenotypes pulled from ontology lookup service (OLS). The aim for this tool is an easy to use interface to view the relevant information gathered from these vast databases. The user can search by gene symbol or phenotype with an autocomplete functionality and from that it will display the phenotype, the gene symbol, the clinical significance (pathogenic, Benign etc..), all the references associated with the entry. The backbone of this tool will be the mysql database where all the information collected from clinvar will be assigned to tables with the main goal for the data to be normalized and less redundant (trying my best 😊). The suggested tables are:

* Genes table which holds gene symbols and hgnc identifiers.
* Variants table which holds variant information(allele id, type of variant, start, stop, cytogenetic location).
* Phenotype table that holds clinical significance information alongside the description of the phenotype.
* Variant\_phenotype will be a table that connects variants to phenotypes.
* A reference table that stores reference ids to link entries to original database (clinvar) or other databases (for transparency and full disclosure).

The initial python code will be responsible for parsing through the information provided by clinvar. The code will filter out any irrelevant entries or entries that have missing components, then using the checked entries to populate the tables mentioned above.

The html can be a simple webpage with a search bar where users can enter a gene symbol or a phenotype to search for. This is going to be straightforward emphasizing on ease of use and not bombarding the user with distracting visual ques. The javascript implementation that I am aiming for will also aid in the ease of use with an auto-completion aspect to suggest matching terms from the database.

The cgi script will act as a bridge between the HTML and the mysql database so when a user performs a search, the cgi script will query the database for matching entries and return results to the frontend. The cgi script will handle incoming requests and html will format and display the data in a readable organized way for the user.

Last proposed aspect of the tool as mentioned above will be the javascript with AJAX calls to enable autcomplete functionality as the user types where the script will communicate with the cgi script to fetch potential matches from the database and display them to the user.