Developing a Consolidated VA Court Case Database

STS 4600

Spring 2021

David Alves

Computer Science

Signature ____________David Alves__________________Date____5/5/20201____

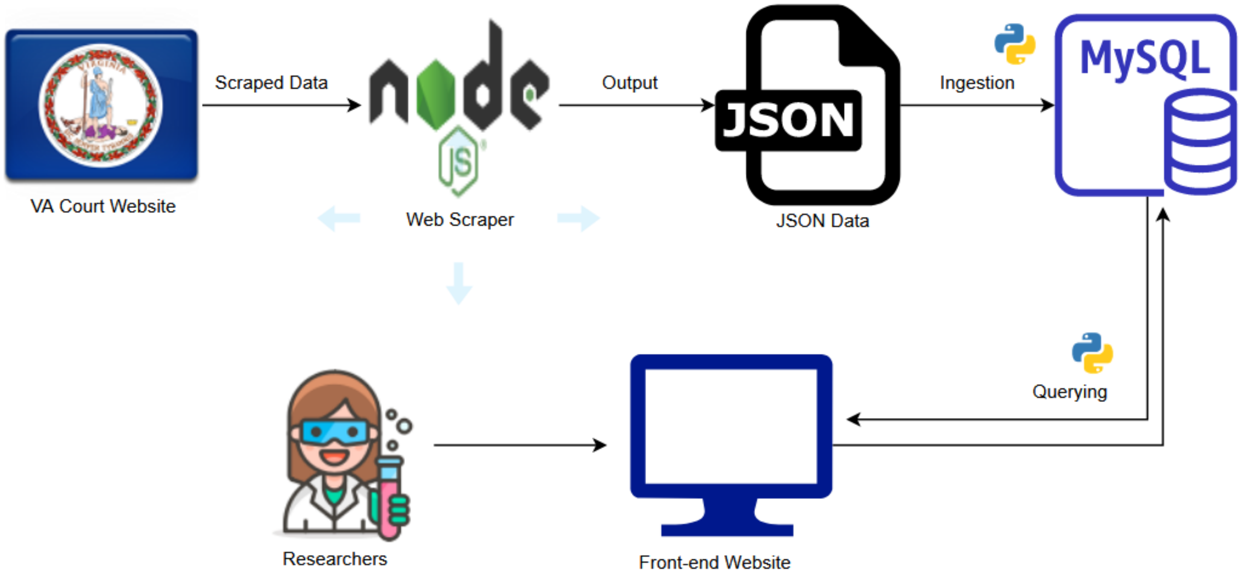
David Alves

Signature ____ Date_____________

Richard D. Jacques

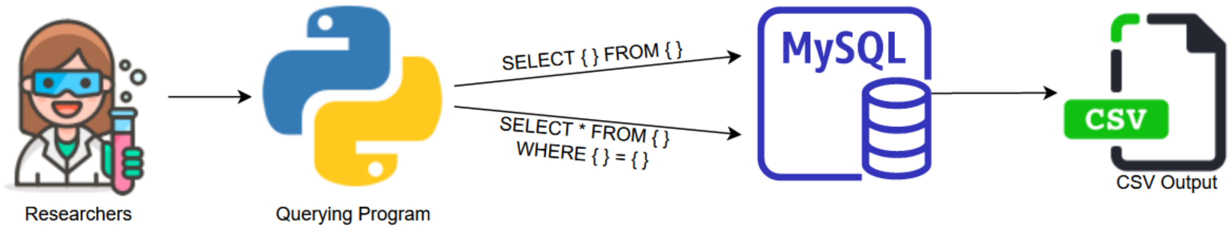
Technical Report

Throughout the course of the semester, this project aimed at making progress towards developing a fully functional web-scraper and web interface for researchers to download the court data contained in the Virginia Online Court Information System (OCIS) website. The overall architecture of this system is as follows: first, a NodeJS web scraper was completed alongside a bash script to ensure that the scraping could be automated. In short, the web scraper contains a captcha solver to unlock the website before effectively pulling the court data and converting it to a JSON format. The JSON court data is then organized into folders by court, and then further separated into criminal and civil cases. Once the data is aggregated, the JSON data is then ingested into a MySQL database for storage and querying capabilities. After the MySQL database is populated, we developed some querying functions in Python to test gathering and converting information stored in our database. We chose csv as the preferred standard of output for our data. Ultimately, the goal was to host this converted data on a frontend website for researchers to query and access. A broad end-to-end overview of our system is shown below.



For my specific contributions, I spent the first couple of weeks setting up my environment and briefly studying the already produced web scraper code left to our team. Because our work was heavily based upon previous work done by Ben Schoenfeld, I then documented the differences between what our web scraper produced and samples of Ben's data. The purpose of this was to ensure that we were scraping at least as much info as Ben's did, and to learn where our scraper could further be developed. I documented that we had we had severe discrepancies between our data, and that our scraper was leaving out a majority of the potential data. At the time, only data under the "Cases" heading was being accumulated, leaving out critically important plaintiff, defendant and hearing information. Regarding the discrepancies, we had various datatype and naming differences between our data and Ben's. This led us to discussions on how we could consolidate and standardize our naming and formatting between the two scrapers. My iteration of this documentation resides on the VA Courts Project Slack page under the #general channel. This document has since been updated by David Stern, who published his updates on the same channel. Thanks to work by Matthew Bacon, the missing information was accounted for and is now being obtained by the newest version of our scraper.

Next, I worked on developing querying functionality for our MySQL database. The querying program had two main objectives. First, we wanted to properly access specific data stored within our MySQL database. This was the actual "querying" functionality. Second, we needed a way to take the queried data and convert it to a standardized format. A comma-separated values (csv) file was chosen as the desired output format, and our program was adjusted to execute the transformation. Upon finishing execution, the querying program is able to handle two different types of queries – shown below – and format the resultant data into an output.csv file. Note that the curly braces, { }, represent values that can be specified by the user.



The first query allows a user to select specific columns from a table. In our case, a column could pertain to fields like name, sex, dob or locality. Should a researcher want a more concise set of data, this query would be useful in filtering fields they deem unnecessary. The second query allows researchers to filter specific entries that satisfy a condition. For example, a researcher might want to access all entries within a specific locality or sex. The flexibility offered by the querying program will hopefully allow researchers to gather only the data that they need. Moreover, because certain fields may not be suitable for public display, these functions will allow us to filter between public and private elements. The final version of my code can be found within the VA Courts server under my user (dra4ae).