WebCAT Error Visualization Dashboard

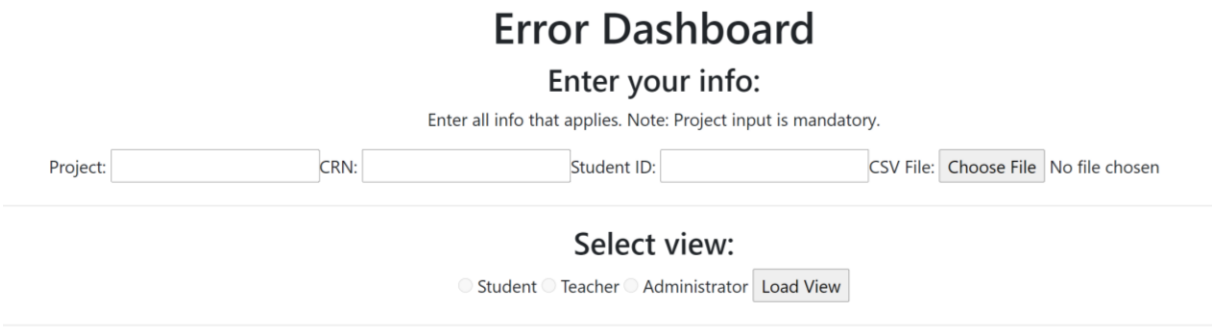
Trevor Miller

How to Use:

1. Load the dashboard.html file in a browser which should be in the same folder as the papaparse files and build-charts.js
2. Enter the csv file (sample file should be in the samplefile folder, for best results, I would recommend looking at 'Lab01' for Project, '12751' for Class, and 's00001' for student because that is the class where I have put the most data into)

Part 1) Form:

- Gathers info about user to decide which charts to generate
- User inputs their own formatted CSV file obtained from WebCAT
- Teacher – Needs to fill in Project and CRN
- Student – Needs to fill in all
- Can switch between views if all relevant info for that view is entered
- Picture below shows what the form looks like:

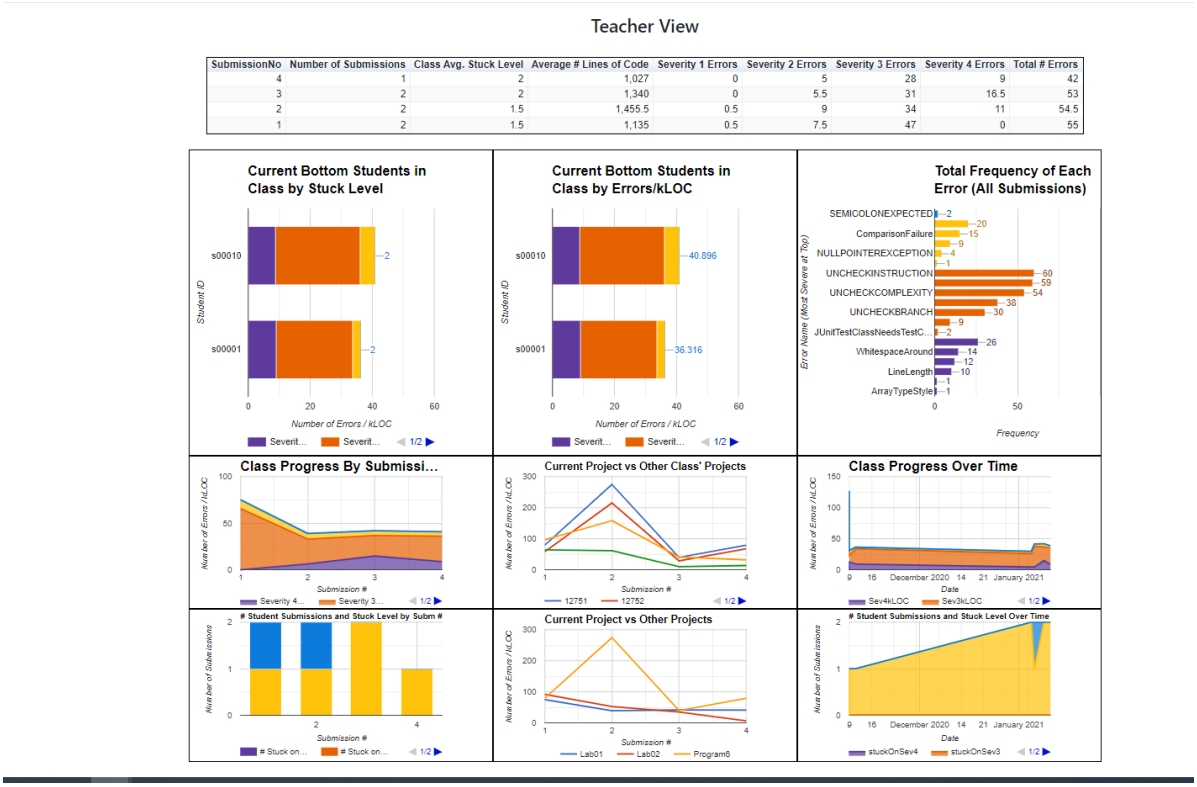


Part 2) Dashboard:

- For most charts, individual errors are assigned a severity 1-4 and each level of severity is displayed on the charts
- Errors per 1000 lines of code is a common measurement used in the visuals
- Charts are either displayed by submission number or over time
- For more info about a certain chart, hover over the data points to see the tooltips
- Student View:
    - Two stacked area charts for how the student is progressing a) over time and b) by submission number
    - Two stacked area charts for how the class is progressing a) over time and b) by submission number

- Bar chart showing the frequency of each error in the student's most recent submission
- Two line charts showing how the student is progressing vs the class 25, 50, 75th percentiles
- Line chart comparing how students are doing compared to their other projects
- Chart showing how many submissions the class has made and their stuck levels on those submissions both over time and by submission number

- Teacher View
  - Stacked bar chart showing the top 20 students with the most errors in terms of number of errors/ kLOC
  - Stacked bar chart showing the top 20 students with the most errors in terms of "stuck level"
  - Bar chart showing the total frequency of each error for all submissions
  - Line charts showing how current project compares to other class' performances on same project and to other projects within the same class
  - Two stacked area charts for how the class is progressing a) over time and b) by submission number
  - Chart showing how many submissions the class has made and their stuck levels on those submissions both over time and by submission number



Part 3) Code Notes

- Once it receives a file to process, it parses the csv file using PapaParse library and generates a google.visualization.DataTable out of the results
    - It is currently programmed to take in the following columns for the csv in the following order: CourseID (string but will usually be a CRN num), Teacher (string), Semester (string), AssignmentNo (string with project name), subjectID (string with subject identifier), SubmissionNo (number), SubmissionTime (string timestamp in the form MM/DD/YY 11:30PM), Lines of Code (number), severity (number 1 – 4), category (string), ClassName (string), FileName (string), MethodName (string), Line (number), Col (number), Error (string)
    - If columns are changed, you will have to update the JSON objects and isColNumeric() function
        - If additional identifiers are added (like semester, teacher, student, etc), the generation functions may have to be updated as well to account for these
- It then creates three more google.visualization.DataTable for different ways of aggregating the DataTable (columns can be found in JSON object at top of .js file)
    - groupedData – groups based on matching courseId, teacher, semester, assignmentNo, subjectID, SubmissionNo
        - Intended to get one row for every student submission to make it easier for visualization

        - Adds a few metrics like number of severity X errors and calculates the number of errors per kLOC
    - groupedClassData - groups based on matching courseId, teacher, semester, assignmentNo, SubmissionNo
        - Intended to get one row for every submission number per class to make it easier for visualization
        - Adds a few metrics like number of severity X errors and calculates the number of errors per kLOC, also adds stuck levels and number of students who submitted
    - groupedClassDataByDate – exact same as groupedClassData except it uses submission time instead of submission number, it is aggregated such that only the most recent submissions are considered
- Then, once the user selects the load view button and either teacher or student view, it will generate the charts for that view