Student Records Manager Documentation

Technologies Used: Html, CSS, JS

Tools Used: VS Code, Live Server Extension

Resources Used: regex101.com for learning about regular expressions, **MDN Web Docs** for refreshing the working of closures.

Brief Summary: This mini project showcases the things we learned including the **ES6+ JS features**, **async/await**, **template literals**, **closures**, and the **principles of input validations**, **responsive design**, and **exception handling** all in one place.

In short we can **Add Students** and each student will have 3 scores, then all the storing, validations, retrievals and displaying of that data will be handled by the app.js files which haves functions that cover the following tasks:

ValidateScoreInput(), this functions performs the validation that only positive integers from 0 to 100 are allowed.

CreateStudentRecord(), this function actually creates the records in the browsers local storage, after all validations are satisfied.

calculateScores(), this function retrieves the records from the local storage and calculates the average, highest and lowest scores and thus displays them accordingly.

After we come to the Student Tables Page by pressing the show *Show All Records* button, the **App.js** here has following functions:

getStudentRecords(), this functions retrieves the records from the local storage and calculates the average and inserts the subsequent table row entries in the student records table.

P.S. I've also used a **closure** here to store the count of the number of records in the local storage which I'm using in the tables' page also.

Challenges Faced: There weren't significant challenges worth mentioning, but I think having to work on the **DOM directly** was very new to me, I had to think different about stuff as I wasn't using react.

The **responsiveness and validation logic** took some time too, but all in all, I don't think this was a challenging task for me.