**Nutrition Services New Project**  
Git Project Location:  
<https://houstonisd@dev.azure.com/houstonisd/HISDNutritionServices/_git/HISDNutritionServices>  
  
Graphical user interface, text, email

Description automatically generated

**React Project  
Text

Description automatically generated**

Where we are so far:  
We have the menu navigation completed  
The only components that have been created are for the Reports  
Student Records Menu Items will be created next  
Graphical user interface, text, application, email

Description automatically generated  
Graphical user interface, text, application, email

Description automatically generated

**WEBAPI WebService**

A screenshot of a computer

Description automatically generated with medium confidence

**GIT LOCATION FOR WEB SERVICE**  
<https://houstonisd@dev.azure.com/houstonisd/CustomHISDWebServices/_git/HISDCustomDevAppsWebAPI>  
  
I used the original WS Project and added the content from my test Sandbox WEBAPI Project

Graphical user interface, text, application, email

Description automatically generated  
  
Graphical user interface

Description automatically generated with low confidence  
  
**I am committing code in VS Code**  
A screenshot of a computer

Description automatically generated with medium confidence  
  
Findings when coding project:

Trying to get a .png file to load with webpack (**local project images**)  
<https://stackoverflow.com/questions/34582405/react-wont-load-local-images>

First run this command:  
npm install --save-dev url-loader file-loader  
  
Then add this to your webpack.config file:  
Text

Description automatically generated  
  
Then reference your image like below:  
Text

Description automatically generated

**Bootstrap Spacing Notation**  
https://getbootstrap.com/docs/4.0/utilities/spacing/

Hiding webpack size limit warnings, just add an entry to your webpack.config

Text

Description automatically generated

**Getting Props to Work with Reusable Controls**  
Along with Reusable Components  
In my project, I was building a search screen. On the search screen, there was a select control that I needed to use multiple times. The only gotcha was, I needed the ID to be different for each select and have a re-usable control.

This was solved by:  
First creating a reusable select control  
Graphical user interface, text, application, chat or text message

Description automatically generated  
Text

Description automatically generated  
  
Text

Description automatically generated  
  
Graphical user interface, text, application

Description automatically generated  
  
And as you can see, I am passing in the props for the ID and it works.  
This was taken from the React Docs  
<https://reactjs.org/docs/components-and-props.html>

This was also proven in my SandboxWebPackWEBApiWire project

Text

Description automatically generated

I got the School Listing Re-Usable Control to work as well  
Graphical user interface, application

Description automatically generated  
**Getting Navigation to work in Reactjs Functional Components**  
<https://www.codegrepper.com/code-examples/javascript/history.push+with+params>  
Text

Description automatically generated  
  
Text

Description automatically generated

And to grab the id

Text

Description automatically generated  
Text

Description automatically generated

Screen Changes:  
Graphical user interface

Description automatically generated  
  
**The Search Screen looks good, just have to wire up the select to navigate to the student detail screen  
Graphical user interface, table

Description automatically generated**

**Getting Bootstrap Icons to work**[**https://www.npmjs.com/package/react-bootstrap-icons**](https://www.npmjs.com/package/react-bootstrap-icons)npm install react-bootstrap-icons --save

**Text

Description automatically generated**

**To get the correct import**

**Graphical user interface, table

Description automatically generated**

import { Pencil  } from 'react-bootstrap-icons';

As you type the name in the { } it will give you a list of icon names to select  
**To see a list of available icons, go to the link below, find the name you want, start typing in the import { }** and it will give you the correct reference  
<https://icons.getbootstrap.com/>

Graphical user interface, application, Teams

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, application, table

Description automatically generated  
  
Getting the URL to work when deployed to IIS  
The new Location:  
http://appsdev.houstonisd.org/NutritionServicesApp/  
You have to have “2” routes for the same component

Diagram

Description automatically generated  
  
Then in your navigation  
Text

Description automatically generated

**Good Video on Passing multiple props to a functional component**

https://www.infoworld.com/article/3603276/how-to-use-react-functional-components.html

[**https://www.youtube.com/watch?v=6aPZR3qAIXg**](https://www.youtube.com/watch?v=6aPZR3qAIXg)

ALTER TABLE tblMilkSub

ALTER COLUMN MilkSubName nvarchar(255);

**Getting dropdown list to work as a functional – reusable component  
In the Administration.js component**You have to use a “call back” to get the onChange event to work

Text

Description automatically generated

You also have to make sure you clear the items from the select as well each time you call the fetch items as well

You also have to use a callBack for the onChange event, notice below in the child component how I am passing the callback handler, **this is because it is an child component**, so for the items as well as the event handler, **you have to pass in a callback handler.**

A screenshot of a computer

Description automatically generated with low confidence

Text

Description automatically generated

**And it works**

**Graphical user interface, application

Description automatically generated**

**I CHANGED THE COE TO WORK WITH A BOOTSTRAP TABLE**

**Graphical user interface, application

Description automatically generated**

**Fixing the modal and race condition when trying to set a value of a modal field before it’s shown**I had situation where I needed to set the value of a input field on a modal popup. The issue was that since the modal was hidden, and I wanted to set the value of the field dynamically, I need to render the modal first in order for the getElementbyID would work. The only way to get it to work was to show the modal first , then pause the thread for a ½ second before setting the value of the field  
A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, application, Teams

Description automatically generated

**Styling the Modal**

https://blog.hubspot.com/website/css-margin-vs-padding#:~:text=In%20CSS%2C%20a%20margin%20is,the%20space%20inside%20an%20element.  
  
In CSS, a margin is the space around an element’s border, while padding is the space between an element’s border and the element’s content. Put another way, the **margin** property controls the space **outside** an element, and the **padding** property controls the space **inside** an element. For example, for a button, the ***text*** **inside** the button’s spacing can be controlled by the **padding**.  
  
The modal needed to have room for long descriptions, and the labels were causing issues, so I nested the labels inside of the same column as the input field. I also added an inline style to make the input longer as well.  
Another thing to note:, you can also have multiple styles as well  
<input id='oldValue' style={{marginLeft: 5, width:300}}>

Graphical user interface, application

Description automatically generated  
Text

Description automatically generated  
  
Checking for an empty string in javascript

Text

Description automatically generated

**Endpoints Tested from service:**

http://localhost:54751/api/Admin/fetchSchoolYears/

http://localhost:54751/api/StudentEntryData/fetchSchoolListings/

http://localhost:54751/api/StudentEntryData/getAllStudentInformationData

http://localhost:54751/api/StudentEntryData/getAllStudentInformationDataTEMPIDS

<http://localhost:54751/api/UploadFiles/getAttachments>

http://localhost:54751/api/StudentEntryData/fetchStudentEntryData/{sqlStatement}

http://localhost:54751/api/StudentEntryData/fetchStudentEntryDataAdminAndSearch/{sqlStatement}

http://localhost:54751/api/StudentEntryData/fetchStudentEntryDataAdminAndSearchLikeClauses/{strFieldValues}

http://localhost:54751/api/Admin/fetchSearchDDListData/ Milk Substitute