Comprehensive review of the select dropdown menu for Programs and Content being offered on Co-x3's Website as well as the star rating display implementation and documentation in respect to user's average rating.

Select dropdown menu: that can be implemented on the review page to select from Programs and Contents being offered. This code is pulling the title for each of the programs and contents from the externalPrograms.json and externalContent.json files and listing them as options in a select dropdown format.

```
import React from 'react';
import programs from './externalPrograms.json';
import content from './externalContent.json';

const DropdownSelect: React.FC = () => {
  return (
    <div>
      <h2>Select a Program</h2>
      <select>
        {programs.map((program, index) => (
          <option key={index} value={program.title}>{program.title}</option>
        ))}
      </select>

      <h2>Select Content</h2>
      <select>
        {content.map((contentItem, index) => (
          <option key={index} value={contentItem.title}>{contentItem.title}</option>
        ))}
      </select>
    </div>
  );
}

export default DropdownSelect;
```

Star Rating Component:

The logic behind the star rating can be seen below. Here we pull the average rating and calculate the number of stars that need to be filled. We fill the stars based on the condition of whether it needs to be filled and if it already has been, then we go to the next condition of partially filling in the star till it meets the percentage of stars that need to be filled. The remaining stars will not be filled. For example 3.7 stars will output 3 stars being filled completely, 70% of star 4 being filled and the last star not being filled.

```
import React from 'react';

interface StarRating{
    averageRating: number;
    totalstars?: number;
}

class StarRating extends React.Component<StarRating, {}>
  {
    render() {
      const { averageRating, totalstars= 5 }= this.props;
      const stars_filled= Math.round(averageRating* totalstars)/ totalstars;
      const starcalc= stars_filled/ totalstars;
      const star_fill_ratio= starcalc* 100;
      const star_roundup= Math.floor(stars_filled);

        let stars= [];
        for(let i= 0; i <totalstars; i++)
        {
        if(i<stars_filled)
            {
                stars.push(<span key={i} className={`star filled`}>⭐</span>);
            }
        else if (i===star_roundup)
            {
                stars.push(<span key={i} className={`star partial`} style={{ width:
`${star_fill_ratio}%` }}>⭐</span>);
            }
        else{
                stars.push(<span key={i} className="star">☆</span>);
            }
        }

        return
        (
        <div>
            {stars} ({averageRating.toFixed(1)})
        </div>
        );
    }
  }
```

Here is the .css file that gives the appearance of the star such as its size and color that will all be outputted inline. Both the .tsx and .css files go hand in hand and there are multiple ways to link these up including having an import statement with a link to the file or using built in libraries.

```css
.star {
  display: inline-block;
  font-size: 24px;
  position: relative;
  overflow: hidden;
  width: 24px;
}

.star.filled {
  color: gold;
}

.star.partial {
  display: inline-block;
  overflow: hidden;
  direction: ltr;
}
```

I also was watching some useful tutorials on how to create a star rating in React. This approach seemed the most efficient way to do it by just importing the star font by installing a package. Looking at our review page that has not been published yet, we already have the code for that but here is a video going into the details of making it. If we need to reimplement it we can use this to update our https://toolbox.co-x3.com/review-us page as well as average rating display above as well.

https://www.youtube.com/watch?v=eDw46GYAIDQ