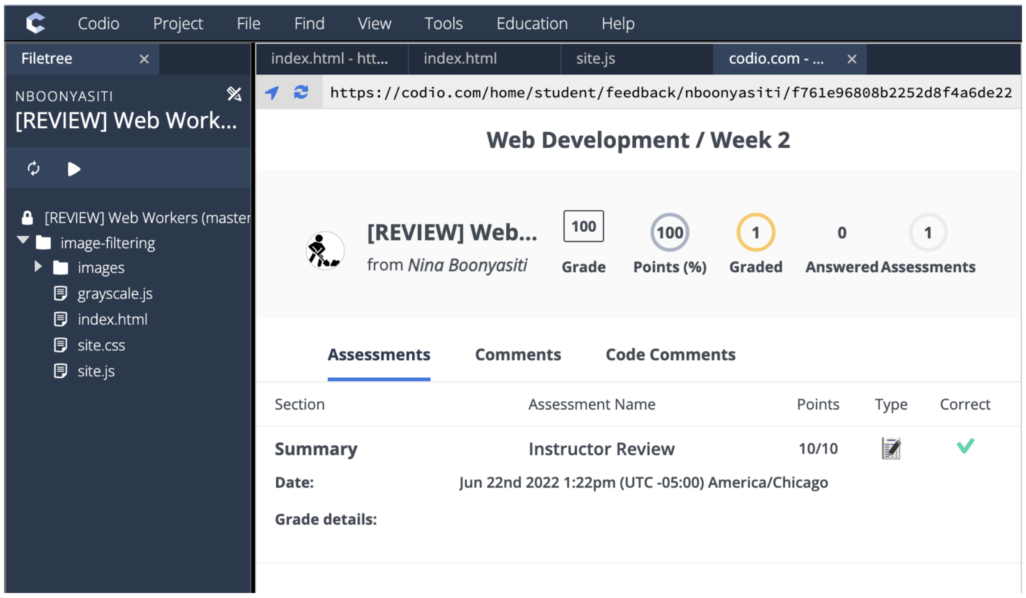
Nina Boonyasiti - 2846101
EECS 692

Assignment: [Tutorial] Web Workers
Due Date: June 17th by 11:59pm
Points: 10
Description: In this tutorial we will use web workers to apply filters to images. You should have an *image-filtering* directory with some starter files. Specifically, the *index.html*, which is pre-loaded with several images from Pixabay.com.

Overall Summary:



https://codio.com/home/student/feedback/nboonyasiti/f761e96808b2252d8f4a6de22

Web Development / Week 2

[REVIEW] Web...
from *Nina Boonyasiti*

100	100	1	0	1
Grade	Points (%)	Graded	Answered	Assessments

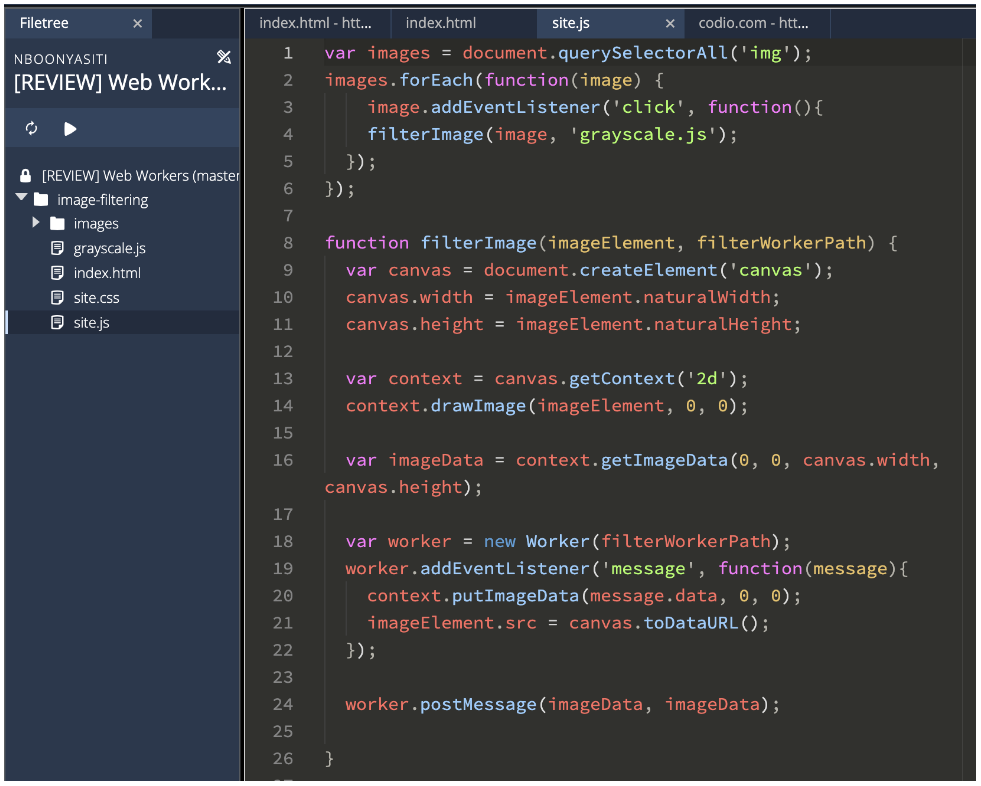
Section	Assessment Name	Points	Type	Correct
Summary	Instructor Review	10/10		✓

Date: Jun 22nd 2022 1:22pm (UTC -05:00) America/Chicago

Grade details:

Nina Boonyasiti - 2846101
EECS 692

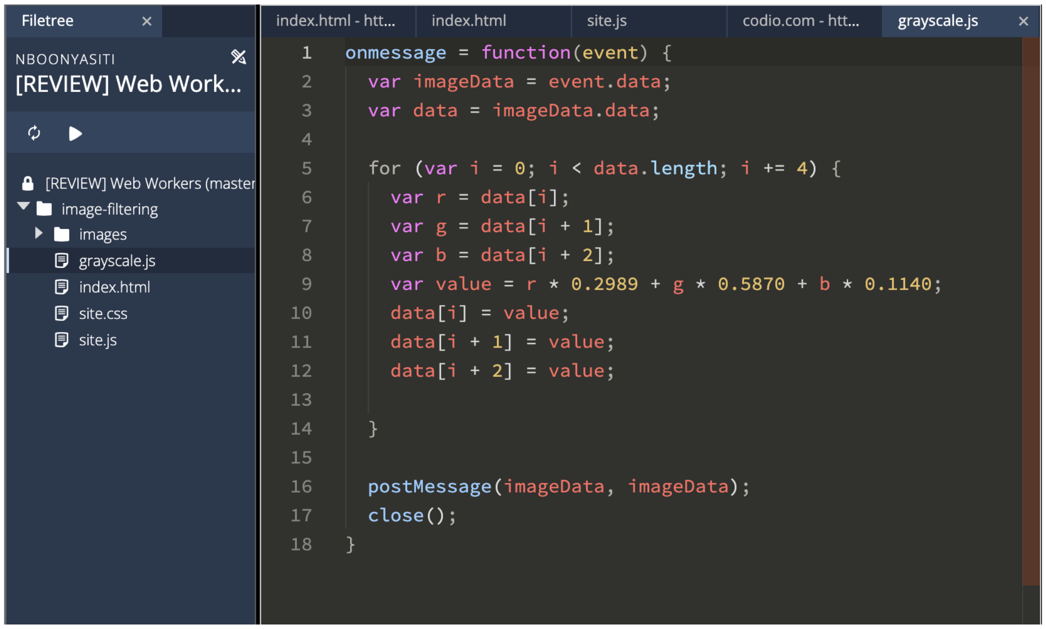
Student Submission:
We first create our filterImage() function to create a “canvas” element and capture the data of the image and save it to our canvas. With this information, we will have a basis of what images we want to select to undergo filtering. Then we have a looping condition to listen for a click event and upon that click event, the image filter is triggered via grayscale.js.



```javascript
var images = document.querySelectorAll('img');
images.forEach(function(image) {
    image.addEventListener('click', function(){
    filterImage(image, 'grayscale.js');
  });
});

function filterImage(imageElement, filterWorkerPath) {
  var canvas = document.createElement('canvas');
  canvas.width = imageElement.naturalWidth;
  canvas.height = imageElement.naturalHeight;

  var context = canvas.getContext('2d');
  context.drawImage(imageElement, 0, 0);

  var imageData = context.getImageData(0, 0, canvas.width,
canvas.height);

  var worker = new Worker(filterWorkerPath);
  worker.addEventListener('message', function(message){
    context.putImageData(message.data, 0, 0);
    imageElement.src = canvas.toDataURL();
  });

  worker.postMessage(imageData, imageData);

}
```

In grayscale.js, we are taking in the image data passed through site.js and iterating each pixel through a for loop. In this loop we want to save each pixel as rgb and change their hexadecimal values to reflect that of a grayscale filter. We save these new values and together they will create a new gray image. After the loop has completed execution, the new image data will be shared back to site.js for index.html to display. A demo of this tutorial is provided in the Week 2 materials.

Nina Boonyasiti - 2846101
EECS 692



```
onmessage = function(event) {
  var imageData = event.data;
  var data = imageData.data;

  for (var i = 0; i < data.length; i += 4) {
    var r = data[i];
    var g = data[i + 1];
    var b = data[i + 2];
    var value = r * 0.2989 + g * 0.5870 + b * 0.1140;
    data[i] = value;
    data[i + 1] = value;
    data[i + 2] = value;

  }

  postMessage(imageData, imageData);
  close();
}
```