Question Results:

2 Code -

```
let possibleColor = ["#5d3fd3", "#a73fd3", "#d33fb5", "#d35d3f", "#d3a73f"];
const irisesWithColors = data.map(iris => {
    return {
        ...iris,
        color: possibleColor[Math.floor(Math.random() * possibleColor.length)]
    };
});
console.log(irisesWithColors);
```

2 Output -

I will put a sample of just three since there's no reason to list all the elements:

```
{
    "sepalLength": 5.1,
    "sepalWidth": 3.5,
    "petalLength": 1.4,
    "petalWidth": 0.2,
    "species": "setosa",
    "color": "#5d3fd3"
}

{
    "sepalLength": 4.9,
    "sepalWidth": 3,
    "petalLength": 1.4,
    "petalWidth": 0.2,
    "species": "setosa",
    "color": "#a73fd3"
}

{
    "sepalLength": 4.7,
    "sepalWidth": 3.2,
    "petalLength": 1.3,
    "petalWidth": 0.2,
    "species": "setosa",
    "color": "#5d3fd3"
}
```

3 Code -

```
const filteredIrises = irisesWithColors.filter(iris => iris.sepalWidth < 4);
    console.log(filteredIrises);
```

3 Output -

147 Objects displayed in console

4 Code -

```
    // Calculate the average petalLength using reduce
    const totalPetalLength = irisesWithColors.reduce((sum, iris) => sum + iris.petalLength, 0);
    const averagePetalLength = totalPetalLength / irisesWithColors.length;
    console.log("Average Petal Length: ", averagePetalLength);
```

4 Output -

Average petal length of 3.7580000000000027 displayed in console

5 Code -

```
   // Find an iris object with petalWidth > 1.0
    const irisWithLargePetalWidth = irisesWithColors.find(iris => iris.petalWidth > 1.0);
    console.log('Iris with petalWidth > 1.0: ', irisWithLargePetalWidth);
```

5 Output -

```
Iris with petalWidth > 1.0:
{
  "sepalLength": 7,
  "sepalWidth": 3.2,
  "petalLength": 4.7,
  "petalWidth": 1.4,
  "species": "versicolor",
  "color": "#d33fb5"
}
```

6 Code -

```
   // Check if any iris object has a petalLength > 10
```

```
 const hasLargePetalLength = irisesWithColors.some(iris => iris.petalLength > 10);
 console.log("Is there any iris with petalLength > 10: ", hasLargePetalLength);
```

6 Output -

```
Is there any iris with petalLength > 10:  false
```

7 Code -

```
// Check if any iris object has a petalLength == to 4.2
 const hasSamePetalLength = irisesWithColors.some(iris => iris.petalLength == 4.2);
 console.log("Is there any iris with petalLength == 4.2: ", hasSamePetalLength);
```

7 Output -

```
Is there any iris with petalLength == 4.2:  true
```

8 Code -

```
// Check if every iris object has a petalWidth < than 3
 const everyLessPetalWidth = irisesWithColors.every(iris => iris.petalWidth < 3);
 console.log("Is every iris with petalWidth < 3: ", everyLessPetalWidth);
```

8 Output -

```
Is every iris with petalWidth < 3:  true
```

9 Code -

```
// Check if every iris object has a petalWidth < than 3
 const everyMoreSepalWidth = irisesWithColors.every(iris => iris.sepalWidth > 1.2);
 console.log("Is every iris with sepalWidth > 1.2: ", everyMoreSepalWidth);
```

9 Output -

```
Is every iris with sepalWidth > 1.2:  true
```

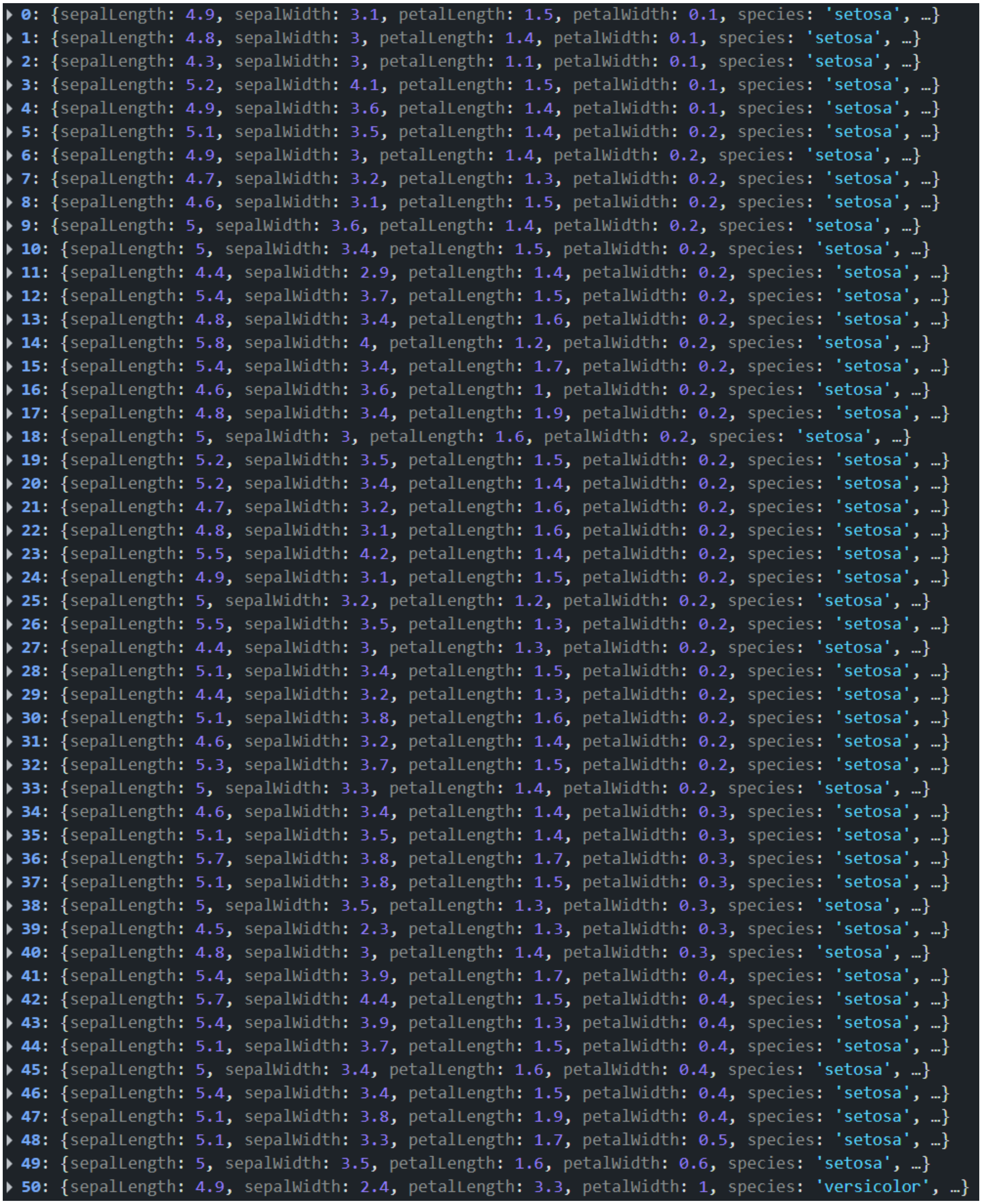
10 Code -

```
// Make an array with all iris objects sorted by petalWidth (lowest to highest)
```

```
        const irisesWithColorsSorted = irisesWithColors.toSorted((a, b) => a.petalWidth -
b.petalWidth);
        console.log(irisesWithColorsSorted);
```

10 Output -

150 Objects, sorted by petalWidth, here's an image of the first 51



```
▸ 0: {sepalLength: 4.9, sepalWidth: 3.1, petalLength: 1.5, petalWidth: 0.1, species: 'setosa', …}
▸ 1: {sepalLength: 4.8, sepalWidth: 3, petalLength: 1.4, petalWidth: 0.1, species: 'setosa', …}
▸ 2: {sepalLength: 4.3, sepalWidth: 3, petalLength: 1.1, petalWidth: 0.1, species: 'setosa', …}
▸ 3: {sepalLength: 5.2, sepalWidth: 4.1, petalLength: 1.5, petalWidth: 0.1, species: 'setosa', …}
▸ 4: {sepalLength: 4.9, sepalWidth: 3.6, petalLength: 1.4, petalWidth: 0.1, species: 'setosa', …}
▸ 5: {sepalLength: 5.1, sepalWidth: 3.5, petalLength: 1.4, petalWidth: 0.2, species: 'setosa', …}
▸ 6: {sepalLength: 4.9, sepalWidth: 3, petalLength: 1.4, petalWidth: 0.2, species: 'setosa', …}
▸ 7: {sepalLength: 4.7, sepalWidth: 3.2, petalLength: 1.3, petalWidth: 0.2, species: 'setosa', …}
▸ 8: {sepalLength: 4.6, sepalWidth: 3.1, petalLength: 1.5, petalWidth: 0.2, species: 'setosa', …}
▸ 9: {sepalLength: 5, sepalWidth: 3.6, petalLength: 1.4, petalWidth: 0.2, species: 'setosa', …}
▸ 10: {sepalLength: 5, sepalWidth: 3.4, petalLength: 1.5, petalWidth: 0.2, species: 'setosa', …}
▸ 11: {sepalLength: 4.4, sepalWidth: 2.9, petalLength: 1.4, petalWidth: 0.2, species: 'setosa', …}
▸ 12: {sepalLength: 5.4, sepalWidth: 3.7, petalLength: 1.5, petalWidth: 0.2, species: 'setosa', …}
▸ 13: {sepalLength: 4.8, sepalWidth: 3.4, petalLength: 1.6, petalWidth: 0.2, species: 'setosa', …}
▸ 14: {sepalLength: 5.8, sepalWidth: 4, petalLength: 1.2, petalWidth: 0.2, species: 'setosa', …}
▸ 15: {sepalLength: 5.4, sepalWidth: 3.4, petalLength: 1.7, petalWidth: 0.2, species: 'setosa', …}
▸ 16: {sepalLength: 4.6, sepalWidth: 3.6, petalLength: 1, petalWidth: 0.2, species: 'setosa', …}
▸ 17: {sepalLength: 4.8, sepalWidth: 3.4, petalLength: 1.9, petalWidth: 0.2, species: 'setosa', …}
▸ 18: {sepalLength: 5, sepalWidth: 3, petalLength: 1.6, petalWidth: 0.2, species: 'setosa', …}
▸ 19: {sepalLength: 5.2, sepalWidth: 3.5, petalLength: 1.5, petalWidth: 0.2, species: 'setosa', …}
▸ 20: {sepalLength: 5.2, sepalWidth: 3.4, petalLength: 1.4, petalWidth: 0.2, species: 'setosa', …}
▸ 21: {sepalLength: 4.7, sepalWidth: 3.2, petalLength: 1.6, petalWidth: 0.2, species: 'setosa', …}
▸ 22: {sepalLength: 4.8, sepalWidth: 3.1, petalLength: 1.6, petalWidth: 0.2, species: 'setosa', …}
▸ 23: {sepalLength: 5.5, sepalWidth: 4.2, petalLength: 1.4, petalWidth: 0.2, species: 'setosa', …}
▸ 24: {sepalLength: 4.9, sepalWidth: 3.1, petalLength: 1.5, petalWidth: 0.2, species: 'setosa', …}
▸ 25: {sepalLength: 5, sepalWidth: 3.2, petalLength: 1.2, petalWidth: 0.2, species: 'setosa', …}
▸ 26: {sepalLength: 5.5, sepalWidth: 3.5, petalLength: 1.3, petalWidth: 0.2, species: 'setosa', …}
▸ 27: {sepalLength: 4.4, sepalWidth: 3, petalLength: 1.3, petalWidth: 0.2, species: 'setosa', …}
▸ 28: {sepalLength: 5.1, sepalWidth: 3.4, petalLength: 1.5, petalWidth: 0.2, species: 'setosa', …}
▸ 29: {sepalLength: 4.4, sepalWidth: 3.2, petalLength: 1.3, petalWidth: 0.2, species: 'setosa', …}
▸ 30: {sepalLength: 5.1, sepalWidth: 3.8, petalLength: 1.6, petalWidth: 0.2, species: 'setosa', …}
▸ 31: {sepalLength: 4.6, sepalWidth: 3.2, petalLength: 1.4, petalWidth: 0.2, species: 'setosa', …}
▸ 32: {sepalLength: 5.3, sepalWidth: 3.7, petalLength: 1.5, petalWidth: 0.2, species: 'setosa', …}
▸ 33: {sepalLength: 5, sepalWidth: 3.3, petalLength: 1.4, petalWidth: 0.2, species: 'setosa', …}
▸ 34: {sepalLength: 4.6, sepalWidth: 3.4, petalLength: 1.4, petalWidth: 0.3, species: 'setosa', …}
▸ 35: {sepalLength: 5.1, sepalWidth: 3.5, petalLength: 1.4, petalWidth: 0.3, species: 'setosa', …}
▸ 36: {sepalLength: 5.7, sepalWidth: 3.8, petalLength: 1.7, petalWidth: 0.3, species: 'setosa', …}
▸ 37: {sepalLength: 5.1, sepalWidth: 3.8, petalLength: 1.5, petalWidth: 0.3, species: 'setosa', …}
▸ 38: {sepalLength: 5, sepalWidth: 3.5, petalLength: 1.3, petalWidth: 0.3, species: 'setosa', …}
▸ 39: {sepalLength: 4.5, sepalWidth: 2.3, petalLength: 1.3, petalWidth: 0.3, species: 'setosa', …}
▸ 40: {sepalLength: 4.8, sepalWidth: 3, petalLength: 1.4, petalWidth: 0.3, species: 'setosa', …}
▸ 41: {sepalLength: 5.4, sepalWidth: 3.9, petalLength: 1.7, petalWidth: 0.4, species: 'setosa', …}
▸ 42: {sepalLength: 5.7, sepalWidth: 4.4, petalLength: 1.5, petalWidth: 0.4, species: 'setosa', …}
▸ 43: {sepalLength: 5.4, sepalWidth: 3.9, petalLength: 1.3, petalWidth: 0.4, species: 'setosa', …}
▸ 44: {sepalLength: 5.1, sepalWidth: 3.7, petalLength: 1.5, petalWidth: 0.4, species: 'setosa', …}
▸ 45: {sepalLength: 5, sepalWidth: 3.4, petalLength: 1.6, petalWidth: 0.4, species: 'setosa', …}
▸ 46: {sepalLength: 5.4, sepalWidth: 3.4, petalLength: 1.5, petalWidth: 0.4, species: 'setosa', …}
▸ 47: {sepalLength: 5.1, sepalWidth: 3.8, petalLength: 1.9, petalWidth: 0.4, species: 'setosa', …}
▸ 48: {sepalLength: 5.1, sepalWidth: 3.3, petalLength: 1.7, petalWidth: 0.5, species: 'setosa', …}
▸ 49: {sepalLength: 5, sepalWidth: 3.5, petalLength: 1.6, petalWidth: 0.6, species: 'setosa', …}
▸ 50: {sepalLength: 4.9, sepalWidth: 2.4, petalLength: 3.3, petalWidth: 1, species: 'versicolor', …}
```

Visualization (11)
Summary of intentions, how we modelled / visualized the data:

It was Kiana’s idea to use eyes for the visualization here, as a clever play on the data being about irises (flowers), as well as to have the interactivity be based on the shutting and opening of the eyes.

While the 5 eyes by themselves display the approximate petalWidth of the data used through the images, which change depending on the index (0 - 4) decided by the petalWidth, I decided it would be nice to also highlight the difference in other perhaps more recognizable ways.

One of the changes I added was the border colors and thickness, which would also change depending on the index of the flower. I think this created a very good visual indicator as to how big the petalWidth was, especially because it is based on the colors of the rainbow, ranging from a more calm cyan to a severe red.

The other big indicator I added was the audio track that only played when an eye was hovered over. I think this connected well with the visual stimuli as its volume was based off of the index as well and added to the cohesiveness of the whole colored border idea.

As for the “sing” audio being chosen as the track as well as the eyes closing, that was really just for decoration and to add interactivity.

The button is there to force the user to click on the page since that is required for audio to play.