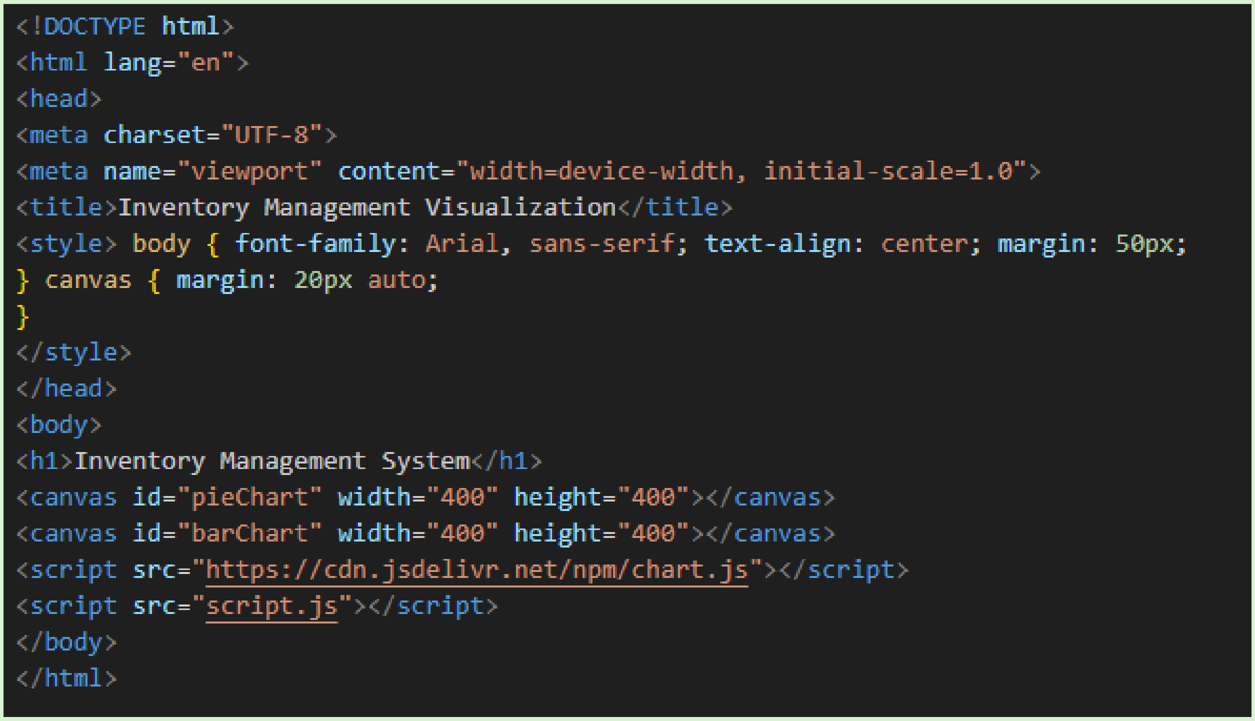
EXPERIMENT-10

AIM : to create data visualizations, such as pie charts and bar graphs, for an inventory management system using JavaScript.

230701204

HTML CODE:



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Inventory Management Visualization</title>
<style> body { font-family: Arial, sans-serif; text-align: center; margin: 50px;
} canvas { margin: 20px auto;
}
</style>
</head>
<body>
<h1>Inventory Management System</h1>
<canvas id="pieChart" width="400" height="400"></canvas>
<canvas id="barChart" width="400" height="400"></canvas>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script src="script.js"></script>
</body>
</html>
```

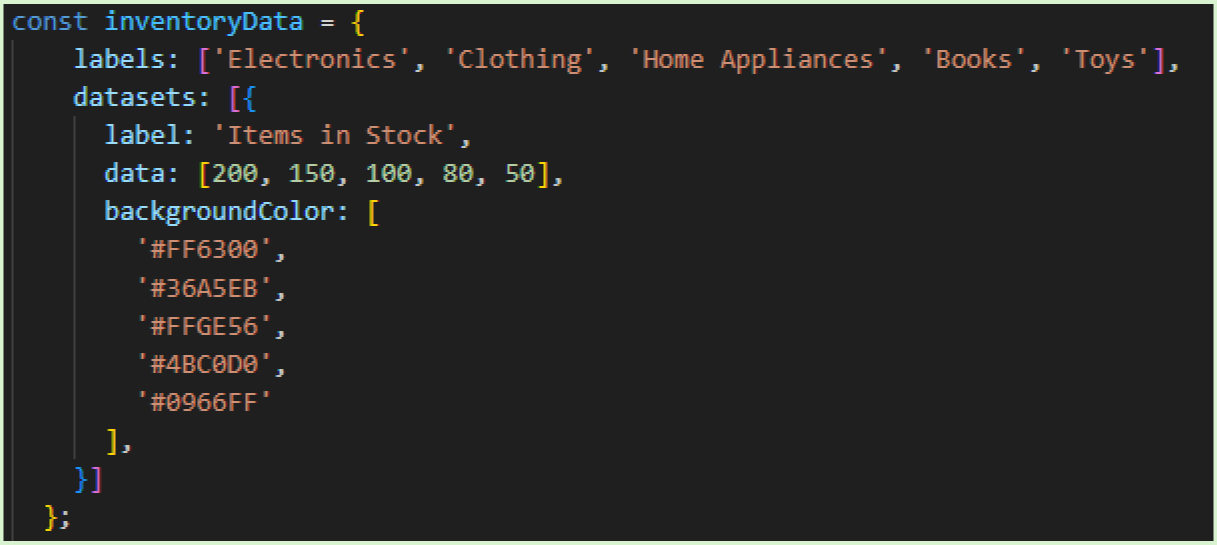
DESCRIPTION:

This is a simple and clean webpage designed to visualize inventory data in a way that’s easy to understand. It starts by setting up the basic page layout—making sure it looks good on all screen sizes and uses a clear font with centered text and generous spacing for a nice visual balance.

Right in the middle of the page, there's a heading that says “Inventory Management System,” followed by two empty canvas areas. Think of these canvases as blank drawing boards—one will display a pie chart and the other a bar chart, both showing how many items are in stock across different product categories.

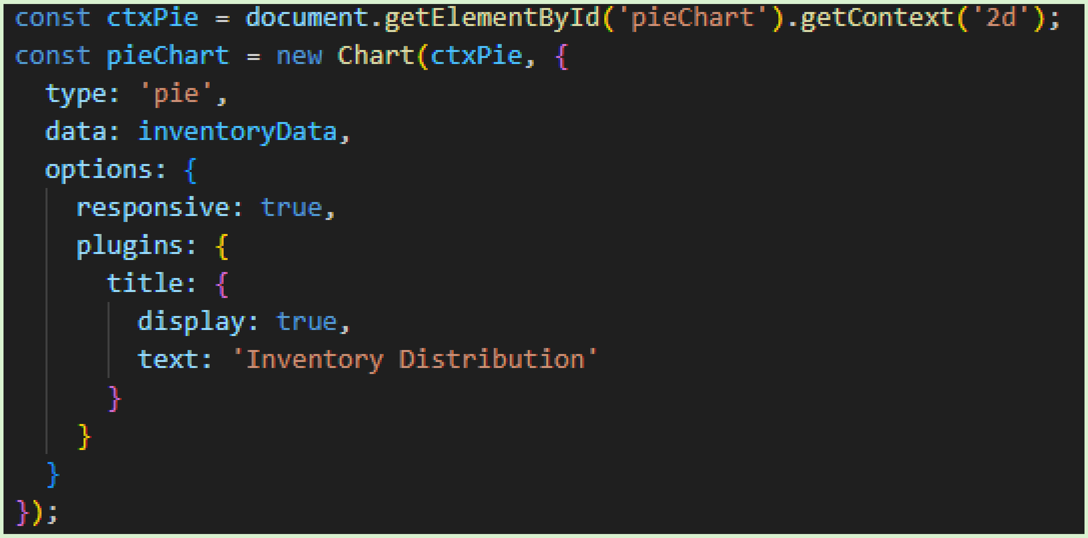
To actually draw those charts, the page brings in Chart.js (a popular JavaScript charting library) from an online source. It also connects to a local script file called script.js, where the real magic happens—this is where the charts get their data, colors, and behavior. It's a great starting point for anyone looking to make their data more visual and interactive.

JS CODE:



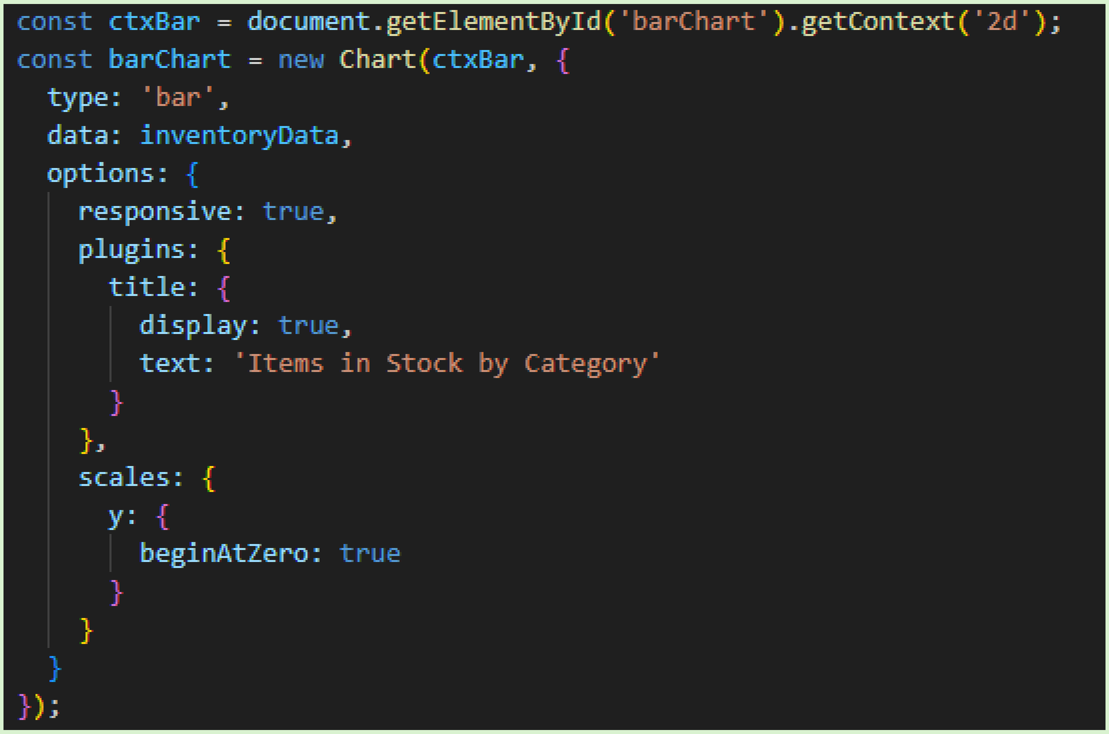
```
const inventoryData = {
    labels: ['Electronics', 'Clothing', 'Home Appliances', 'Books', 'Toys'],
    datasets: [{
      label: 'Items in Stock',
      data: [200, 150, 100, 80, 50],
      backgroundColor: [
        '#FF6300',
        '#36A5EB',
        '#FFGE56',
        '#4BC0D0',
        '#0966FF'
      ],
    }]
  };
```

This code creates the data behind the charts. It lists product categories like Electronics and Toys, and pairs them with the number of items in stock. Each category is also given a unique color to make the charts visually clear and vibrant. It's the core dataset powering your visuals.



```
const ctxPie = document.getElementById('pieChart').getContext('2d');
const pieChart = new Chart(ctxPie, {
  type: 'pie',
  data: inventoryData,
  options: {
    responsive: true,
    plugins: {
      title: {
        display: true,
        text: 'Inventory Distribution'
      }
    }
  }
});
```

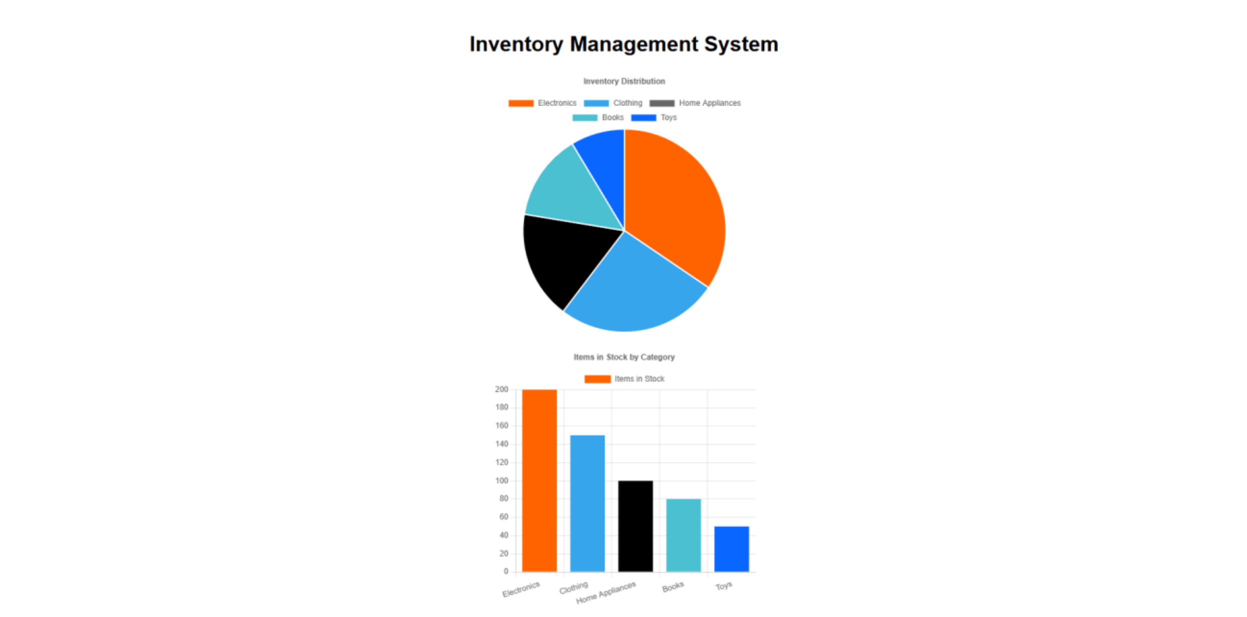
This part draws a pie chart on the page. It connects to the canvas element labeled pieChart, uses the inventory data, and displays it in a responsive, round format. The title “Inventory Distribution” appears above the chart, helping users quickly grasp how stock is divided across categories.



```
const ctxBar = document.getElementById('barChart').getContext('2d');
const barChart = new Chart(ctxBar, {
  type: 'bar',
  data: inventoryData,
  options: {
    responsive: true,
    plugins: {
      title: {
        display: true,
        text: 'Items in Stock by Category'
      }
    },
    scales: {
      y: {
        beginAtZero: true
      }
    }
  }
});
```

This block sets up a bar chart using the same inventory data. It targets the bar chart canvas and shows the data in vertical bars, making it easy to compare quantities. The chart starts at zero and includes a title: “Items in Stock by Category” for clear and immediate understanding.

OUTPUT:



Inventory Management System

This image displays the final visual output of your inventory management system using Chart.js. At the top, a pie chart clearly illustrates how different product categories—like Electronics and Books—contribute to the overall stock. Below it, a bar chart gives a side-by-side view of the exact quantity in each category, making comparisons easy.