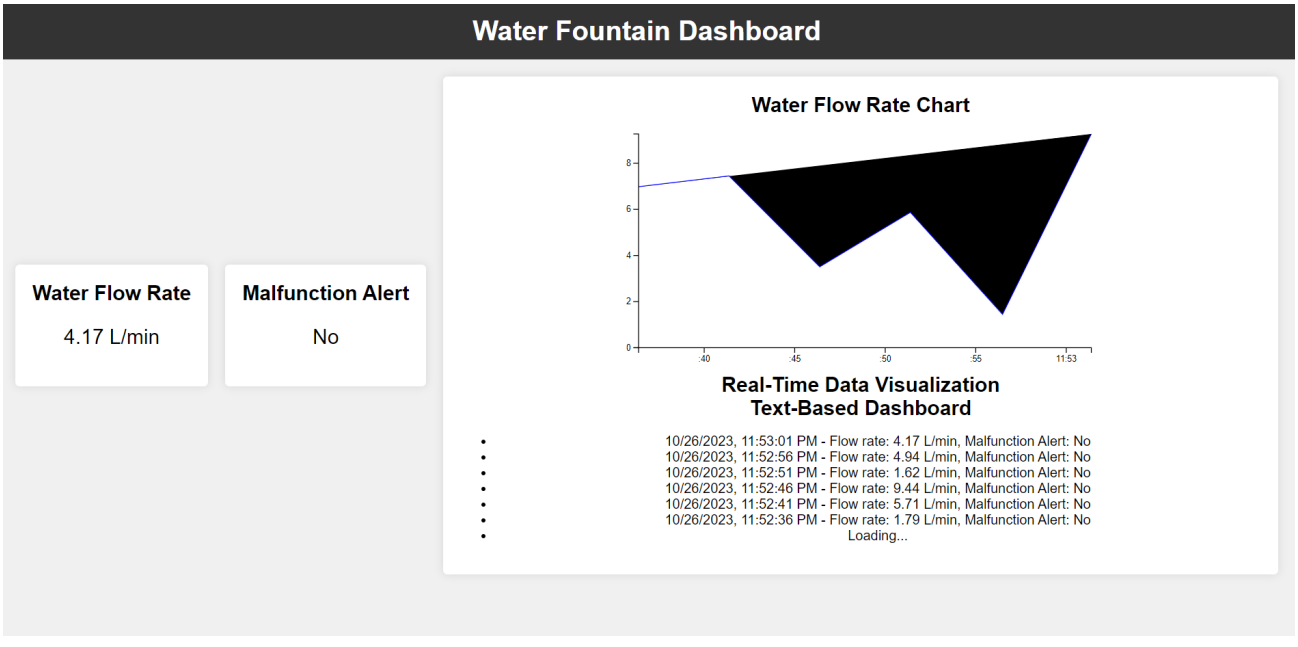
Smart Water Fountains

Web Page for Real Time Monitoring



HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" href="style.css">

  <script src="https://d3js.org/d3.v7.min.js"></script> <!-- Include D3.js -->

  <title>Water Fountain Dashboard</title>

</head>
```

```
<body>
  <header>
    <h1>Water Fountain Dashboard</h1>
  </header>
  <main>
    <div class="status">
      <h2>Water Flow Rate</h2>
      <p id="flow-rate">Loading...</p>
    </div>
    <div class="status">
      <h2>Malfunction Alert</h2>
      <p id="malfunction-alert">Loading...</p>
    </div>
    <div class="dashboard">
        <h2>Water Flow Rate Chart</h2>
        <div id="line-chart"></div>
      <h2>Real-Time Data Visualization</h2>
      <div id="bar-chart"></div>
      <h2>Text-Based Dashboard</h2>
      <ul id="event-list">
        <li>Loading...</li>
      </ul>
    </div>
  </main>
  <script src="script.js"></script>
</body>
</html>
```

CSS

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
    text-align: center;
}
header {
    background-color: #333;
    color: #fff;
    padding: 15px;
}
h1 {
    margin: 0;
}
main {
    display: flex;
    justify-content: space-between;
    align-items: center;
    flex-wrap: wrap;
    gap: 20px;
    margin: 20px;
}
.status {
    background-color: #fff;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
```

```
    text-align: center;
}
.dashboard {
    flex: 1;
    padding: 20px;
    background-color: #fff;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    text-align: center;
}
#line-chart {
    height: 300px;
}
h2 {
    margin: 0;
}
p {
    font-size: 24px;
}
```

JavaScipt

```
// Define global variables for storing historical data
const flowRateData = [];
// Create the line chart using D3.js
const margin = { top: 20, right: 30, bottom: 30, left: 40 };
const width = 600 - margin.left - margin.right;
const height = 300 - margin.top - margin.bottom;
const svg = d3.select("#line-chart")
```

```
    .append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
    .append("g")
    .attr("transform", `translate(${margin.left},${margin.top})`);
const xScale = d3.scaleTime().range([0, width]);
const yScale = d3.scaleLinear().range([height, 0]);
const line = d3.line()
    .x((d) => xScale(d.time))
    .y((d) => yScale(d.flowRate));
svg.append("path")
    .attr("class", "line")
    .style("stroke", "blue");
function updateTextDashboard() {
    // Simulate fetching data from an API
    const flowRate = (Math.random() * 10).toFixed(2);
    const malfunction = Math.random() > 0.9;
    const timestamp = new Date().toLocaleString();
    // Update the DOM elements in the text-based dashboard
    document.getElementById("flow-rate").textContent = flowRate + " L/min";
    document.getElementById("malfunction-alert").textContent = malfunction ? "Yes" : "No";
    // Update recent events
    const eventList = document.getElementById("event-list");
    const eventItem = document.createElement("li");
    eventItem.textContent = `${timestamp} - Flow rate: ${flowRate} L/min, Malfunction Alert: ${malfunction ?
"Yes" : "No"}`;
    eventList.insertBefore(eventItem, eventList.firstChild);
}// Update status with a line chart
```

```
function updateLineChart() {
    // Simulate fetching data from an API
  const flowRate = (Math.random() * 10).toFixed(2);
  const timestamp = new Date();
  // Add the new data point to the dataset
  flowRateData.push({ time: timestamp, flowRate: flowRate });
  // Keep a fixed number of data points for the chart (e.g., 30 points)
  if (flowRateData.length > 30) {
    flowRateData.shift();
  }
  // Update the x and y domains for the chart
  xScale.domain(d3.extent(flowRateData, (d) => d.time));
  yScale.domain([0, d3.max(flowRateData, (d) => d.flowRate)]);
  // Update the line path
  svg.select(".line")
    .data([flowRateData])
    .attr("d", line);
  // Update the x-axis
  svg.select(".x-axis")
    .call(d3.axisBottom(xScale));
  // Update the y-axis
  svg.select(".y-axis")
    .call(d3.axisLeft(yScale).ticks(5));
  // Update the DOM elements
  document.getElementById("flow-rate").textContent = flowRate + " L/min";
  document.getElementById("malfunction-alert").textContent = Math.random() > 0.9 ? "Yes" : "No";
}
// Add x and y axes to the chart
```

```
svg.append("g")
    .attr("class", "x-axis")
    .attr("transform", `translate(0, ${height})`)
    .call(d3.axisBottom(xScale));
svg.append("g")
    .attr("class", "y-axis")
    .call(d3.axisLeft(yScale).ticks(5));
function updateStatus() {
    // Call the functions to update the line chart and text-based dashboard
    updateLineChart();
    updateTextDashboard();
}// Update status every 5 seconds (5000 milliseconds)
setInterval(updateStatus, 5000);
// Initial update
updateStatus();
```

Team Members:

Jasper Daniel C

Dhanush Raja R

Mohamed Sheik M

Lingam S

Mentor:

Dr.M.Parameswari