Dashboard for Healthcare Data Cleaning

```python
import webbrowser

import json

from http.server import HTTPServer, SimpleHTTPRequestHandler

import threading


# Step 1: Create an HTML file with a built-in data dashboard

html_content = """

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Healthcare Data Cleansing Dashboard</title>

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

</head>

<body>

    <div class="container mt-5">

        <h1 class="text-center">Healthcare Data Cleansing Dashboard</h1>


        <div class="row mt-4">
```

```html
<!-- Data Quality Score -->

<div class="col-md-6">

  <div class="card">

    <div class="card-header">Overall Data Quality Score</div>

    <div class="card-body text-center">

      <div class="rounded-circle bg-light mx-auto" style="width: 150px; height: 150px; display: flex; align-items: center; justify-content: center;">

        <h3 class="text-success" id="quality-score">85%</h3>

      </div>

      <p>Higher score indicates cleaner healthcare data.</p>

    </div>

  </div>

</div>


<!-- Anomaly Detection Chart -->

<div class="col-md-6">

  <div class="card">

    <div class="card-header">Anomalous Records</div>

    <div class="card-body">

      <canvas id="anomalyChart" width="400" height="200"></canvas>

    </div>

  </div>

</div>
```

```html
        </div>


<div class="row mt-4">

    <!-- Data Table -->

    <div class="col-md-6">

        <div class="card">

            <div class="card-header">Healthcare Records</div>

            <div class="card-body">

                <table class="table table-striped" id="records-table">

                    <thead>

                        <tr>

                            <th>Record ID</th>

                            <th>Patient Name</th>

                            <th>Status</th>

                            <th>Action</th>

                        </tr>

                    </thead>

                    <tbody>

                        <!-- Data will be inserted here -->

                    </tbody>

                </table>

            </div>
```

```html
        </div>

      </div>


      <!-- Quick Actions -->

      <div class="col-md-6">

        <div class="card">

          <div class="card-header">Quick Actions</div>

          <div class="card-body">

            <h5>Flagged Records</h5>

            <ul id="flagged-records" class="list-group mb-3">

              <li class="list-group-item">No flagged records yet</li>

            </ul>

            <button class="btn btn-danger w-100 mb-2"
onclick="flagAllAnomalies()">Flag All Anomalies</button>

            <button class="btn btn-warning w-100 mb-2"
onclick="notifyAdmins()">Notify Admins</button>

            <button class="btn btn-primary w-100" onclick="exportReport()">Export
Report</button>

          </div>

        </div>

      </div>

    </div>

  </div>
```

```html
<!-- JS Libraries -->

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script>

   // Sample Healthcare Data (Preloaded)

   let healthcareRecords = [

      { id: 1, name: "John Doe", status: "Clean" },

      { id: 2, name: "Jane Smith", status: "Anomalous" },

      { id: 3, name: "Michael Brown", status: "Clean" },

      { id: 4, name: "Alice Johnson", status: "Anomalous" }

   ];


   // Function to Load Data into the Table

   function loadRecords() {

      const tableBody = document.querySelector("#records-table tbody");

      const flaggedList = document.getElementById("flagged-records");

      tableBody.innerHTML = "";

      flaggedList.innerHTML = "";


      healthcareRecords.forEach(record => {

         const row = `

            <tr>

               <td>${record.id}</td>
```

```
            <td>${record.name}</td>

                <td class="${record.status === 'Anomalous' ? 'text-danger' : 'text-
success'}">${record.status}</td>

                <td><button class="btn btn-sm btn-outline-danger"
onclick="flagRecord(${record.id})">Flag</button></td>

            </tr>

        `;

        tableBody.insertAdjacentHTML("beforeend", row);

        if (record.status === "Anomalous") {

            flaggedList.insertAdjacentHTML("beforeend", <li class="list-group-
item">Record ${record.id}</li>);

        }

    });



    if (flaggedList.children.length === 0) {

        flaggedList.innerHTML = '<li class="list-group-item">No flagged records yet</li>';

    }

}



    // Create Chart

    function createAnomalyChart() {

        const ctx = document.getElementById('anomalyChart').getContext('2d');

        new Chart(ctx, {

            type: 'doughnut',
```

```
    data: {

      labels: ['Clean', 'Anomalous'],

      datasets: [{

        data: [2, 2], // Example counts

        backgroundColor: ['#28a745', '#dc3545']

      }]

    },

    options: {

      responsive: true,

      plugins: {

        legend: {

          position: 'bottom'

        }

      }

    }

  });

}


// Quick Actions

function flagAllAnomalies() {

  alert('Flagging all anomalous records!');

}
```

```
        function notifyAdmins() {

            alert('Admins have been notified!');

        }

        function exportReport() {

            alert('Exporting report...');

        }

        function flagRecord(recordId) {

            alert(Record ${recordId} flagged for review!);

        }


        // Initialize Dashboard

        function initDashboard() {

            loadRecords();

            createAnomalyChart();

        }

        initDashboard();

    </script>

</body>

</html>

"""
```

Step 2: Save the HTML file

```python
with open("dashboard.html", "w") as file:

    file.write(html_content)


# Step 3: Start a Local Server & Open Dashboard

def start_server():

    PORT = 8000

    server_address = ("", PORT)

    httpd = HTTPServer(server_address, SimpleHTTPRequestHandler)

    print(f"🚀 Serving dashboard at: http://localhost:{PORT}/dashboard.html")

    httpd.serve_forever()


# Run server in a separate thread

threading.Thread(target=start_server, daemon=True).start()


# Open in browser

webbrowser.open("http://localhost:8000/dashboard.html")
```