Health Monitor

Task

Design and development of a Health Monitor dashboard displaying data of services running on a machine.

Components

1. Fetching the service data from the machine and pushing it into a database
2. **Accessing the database and making the data available for display on the dashboard.**
3. Displaying the data in a tabular form on the dashboard (Web/Mobile).

Technologies Used

- C# for writing the getServices API
- MySQL DBMS
- HTML/CSS with bootstrap for designing the dashboard.
- PHP for extracting the data from the database.
- JavaScript and JQuery to handle webpage/app events

Task Description

For establishing the connection to the MySQL database, MySQLi extension of PHP was used. **mysqli_connect()** function was used for establishing the connection. The target table was queried using the mysqli_query(). A simple query which returns all the tuples of the tables was executed and the results were stored in an array. The array was then echoed in form of a JSON object.

Code for establishing the connection:

```
<?php

header("Access-Control-Allow-Origin: *");

$con = mysqli_connect("localhost","root","","team") or die ("could not connect
database");

?>
```

Code for querying the target table:

```
<?php

include "db.php";

$data=array();

$name = array();$q=mysqli_query($con,"select * from healthdata");
```

```
while ($row=mysqli_fetch_object($q)){
       $data[]=$row;
       $name[] = $row->Service_Name;
}
$fp = fopen('names.json', 'w');
fwrite($fp, json_encode($name));
fclose($fp);
echo json_encode($data);
?>
```

The query was modified to fetch different kinds of results and fields.

On the HTML end, the data echoed in JSON object was extracted using getJSON() method. Looping over the result set, the attributes of the tuples were stored in local variables and added to the table tag in form of a row (<tr>…</tr>). The function setInterval() was used to run the script at an interval of 2 seconds to make the data dynamic and reflect the changes that occur in the database in real time. Extra details about a particular service were displayed in a modal.

Code Snippet:

```
    $(document).ready(setInterval(function() {
        var url = "http://localhost/scripts/json.php";

        $.getJSON(url, function(result) {
            var tr;
            $('#td').html('');
            console.log(result);
            $.each(result, function(i, field) {
                tr = $('<tr/>');
                tr = $('<tr/>');
               var name = field.Service_Name;
                var status = field.Status;
                var type = field.Service_Type;
                var time = field.TIME;
                var dname = field.Display_Name;
                var csd = field.Can_Shut_Down;
                var cs = field.Can_Stop;
                var cpc = field.Can_Pause_Continue;
                var sdo = field.Services_Depended_On;
                tr.append("<td>" + name + "</td>");

             if(status=="Running")
                   tr.append("<td><span class='glyphicon glyphicon-ok'
style='color:rgb(72,181,163); font-size: 25px;'>" + "</span></td>");
```

```
            else
                  tr.append("<td> <span class='glyphicon glyphicon-remove'
style='color:red; font-size: 25px;'>"  + "</span></td>");
                  tr.append("<td>" + dname + "</td>");
                  tr.append("<td>" + type + "</td>");
                  tr.append("<td>" + time + "</td>");
                  tr.append("<td><button  id = '"+name+"type='button' data-
            toggle='modal' data-target='#myModal1'
            onClick='myfunc(\""+name+"\")'>Info</button></td>");
                $('#td').append(tr);
            });
        });
```

Flow Diagram