https://www.halvorsen.blog

Plotting Data using Google Charts

Hans-Petter Halvorsen

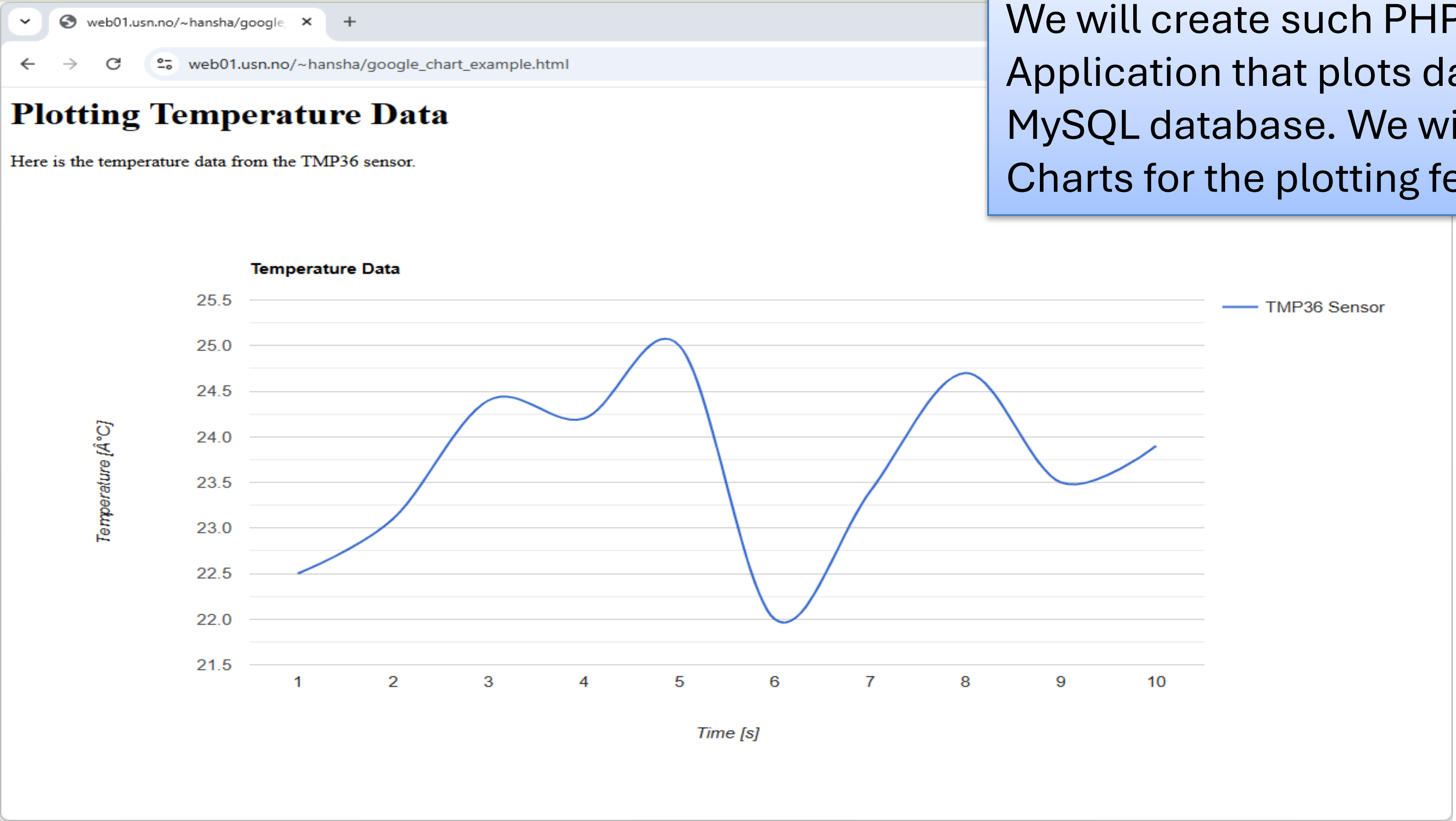
Contents

- Introduction
- Google Charts
- Database (MySQL and phpMyAdmin tool)
- Retrieving Data (PHP server-side)
- Plotting Data from Database (Using Google Charts on Client-side)
- Improvements

Introduction

Google Charts Example

We will create such PHP Web Application that plots data stored in a MySQL database. We will use Google Charts for the plotting features.



web01.usn.no/~hansha/google
web01.usn.no/~hansha/google_chart_example.html

Plotting Temperature Data

Here is the temperature data from the TMP36 sensor.

Introduction

- The purpose with this tutorial is to demonstrate how we can create plots, charts and diagram when creating PHP Web Applications.
- PHP has no built-in functionality for creating charts and plots.
- In this tutorial we will use Google Charts for that purpose.
- We will use MySQL for data storage and use PHP to retrieve data from the database (server-side). Then we will use Google Charts on the client-side to create plots/charts based on the data from the database.
- The focus is to show the basic principles, while code quality and robustness, etc. is not in focus in this tutorial.

Tools

- **PHP** - a server scripting language for making dynamic web pages, typically communicating with a Database.
- We will host our PHP files on an existing **Web Server** that supports PHP and MySQL. You can also create your own or use an existing hosting provider.
- We will use **Visual Studio Code** (you can use another IDE if you prefer).
- We will transfer the local files to the Web Server using **FTP** (File Transfer Protocol). We will use **WinSCP** (you can use another FTP tool if you prefer).
- **MySQL** - a widely used relational database management system (RDBMS). MySQL is free and open-source.
- **phpMyAdmin** - a free and open-source administration tool for MySQL (and MariaDB).
- **Google Charts** – a free chart library (client-side) that can be used to show plots, charts and diagrams on web pages.

Google Charts

Table of Contents

Google Charts

- There exists many different libraries, APIs or frameworks for making charts and plots for your web pages, these are typically using JavaScript and are implemented client-side.
- Google Charts is an API (or framework) for creating Charts in your web pages.
- It is free to use.
- It is a client-side framework/API.
- It is easy to use (when you first know how to use it).
- Google Charts offers many different types of charts: Line Chart, Bar Chart, Column Chart, Pie Chart, etc.
- You can get a detailed overview here: https://developers.google.com/chart

How to implement Google Charts

The most common way to use Google Charts is with simple JavaScript that you embed in your web page.

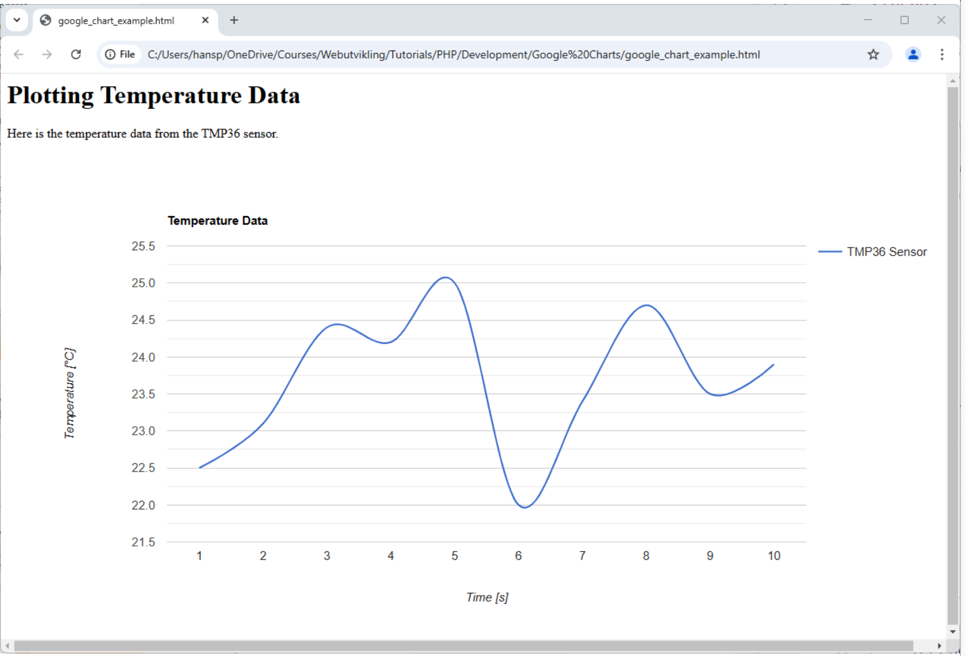
1. Load the Google Chart Libraries.
2. List the Data to be charted.
3. Select Options to customize your chart.
4. Create a Chart Object with an id that you choose.
5. Display: Create a <div> tag with that id to display the Google Chart.

Google Chart libraries

First, you need to load the Google Chart libraries into your webpage:

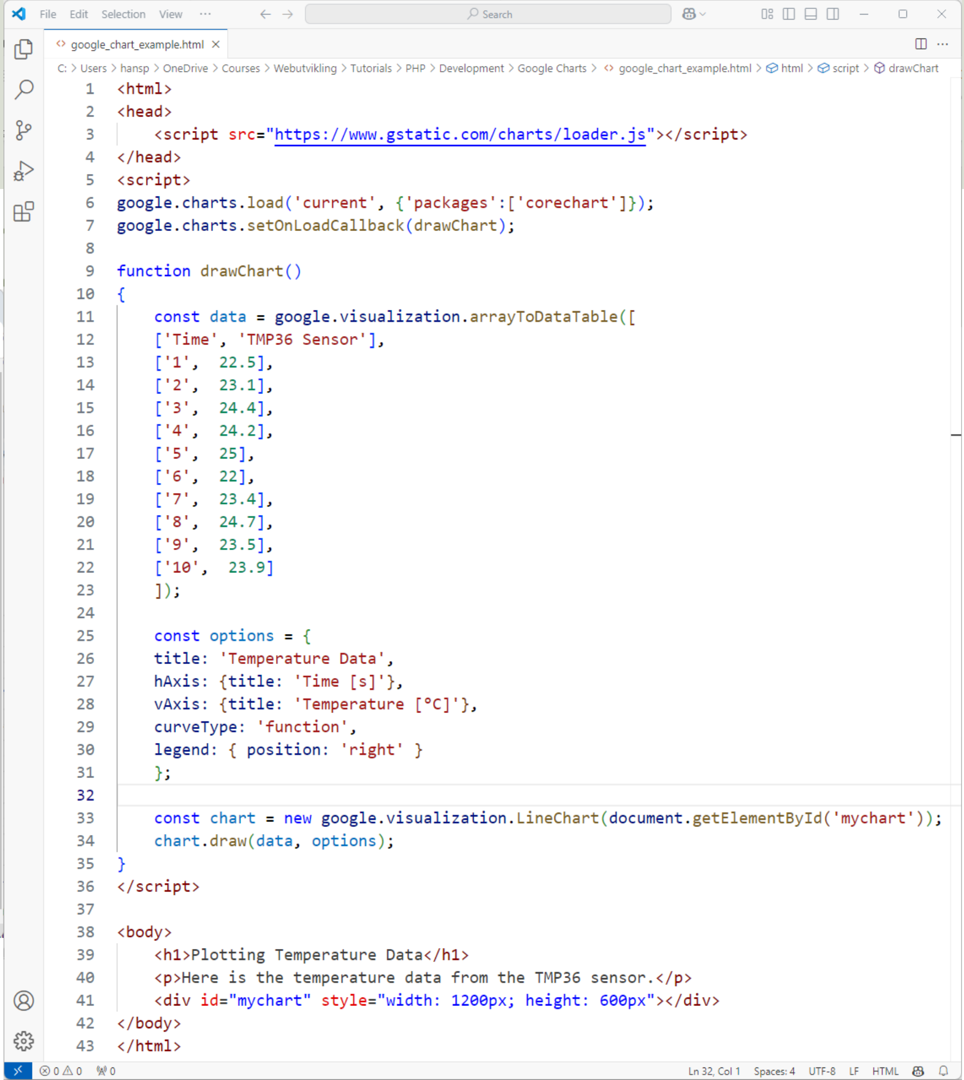
```
<script src="https://www.gstatic.com/charts/loader.js"></script>

<script>
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawChart);
</script>
```

Plot Example


google_chart_example.html
File
C:/Users/hansp/OneDrive/Courses/Webutvikling/Tutorials/PHP/Development/Google%20Charts/google_chart_example.html
Temperature Data
25.5
25.0
24.5
24.0
23.5
23.0
22.5
22.0
21.5
Temperature [°C]
1
2
3
4
5
6
7
8
9
10
Time [s]
TMP36 Sensor

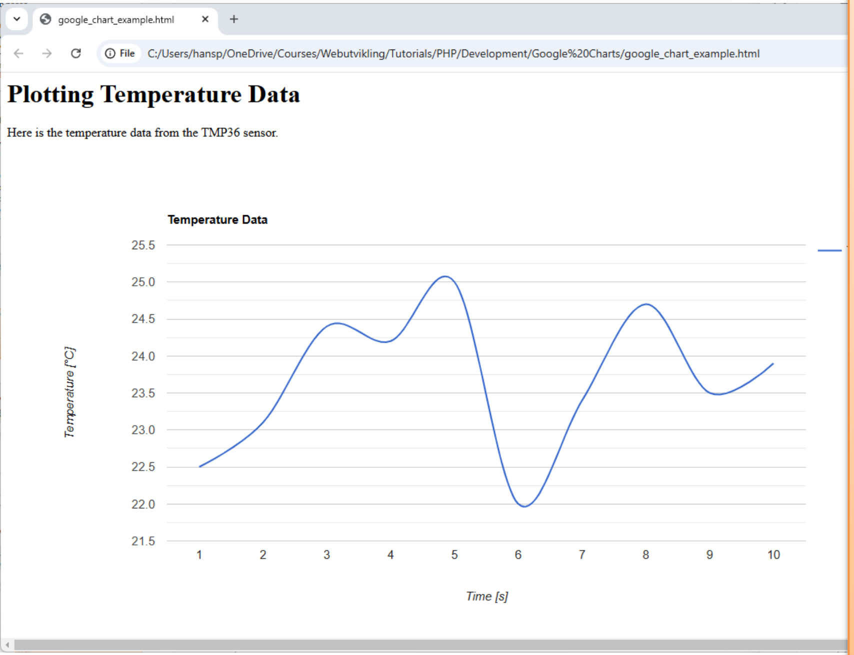
Plotting Temperature Data

Here is the temperature data from the TMP36 sensor.


File
Edit
Selection
View
Search
google_chart_example.html
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Google Charts > google_chart_example.html > html > script > drawChart
Ln 32, Col 1
Spaces: 4
UTF-8
LF
HTML

```
<html>
<head>
    <script src="https://www.gstatic.com/charts/loader.js"></script>
</head>
<script>
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawChart);

function drawChart()
{
    const data = google.visualization.arrayToDataTable([
    ['Time', 'TMP36 Sensor'],
    ['1',  22.5],
    ['2',  23.1],
    ['3',  24.4],
    ['4',  24.2],
    ['5',  25],
    ['6',  22],
    ['7',  23.4],
    ['8',  24.7],
    ['9',  23.5],
    ['10',  23.9]
    ]);

    const options = {
    title: 'Temperature Data',
    hAxis: {title: 'Time [s]'},
    vAxis: {title: 'Temperature [°C]'},
    curveType: 'function',
    legend: { position: 'right' }
    };

    const chart = new google.visualization.LineChart(document.getElementById('mychart'));
    chart.draw(data, options);
}
</script>

<body>
    <h1>Plotting Temperature Data</h1>
    <p>Here is the temperature data from the TMP36 sensor.</p>
    <div id="mychart" style="width: 1200px; height: 600px"></div>
</body>
</html>
```

Code Listing



google_chart_example.html
File
C:/Users/hansp/OneDrive/Courses/Webutvikling/Tutorials/PHP/Development/Google%20Charts/google_chart_example.html

Plotting Temperature Data

Here is the temperature data from the TMP36 sensor.

Temperature Data
25.5
25.0
24.5
24.0
23.5
23.0
22.5
22.0
21.5
Temperature [°C]
1
2
3
4
5
6
7
8
9
10
Time [s]

```
<html>
<head>
    <script src="https://www.gstatic.com/charts/loader.js"></script>
</head>
<script>
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawChart);

function drawChart()
{
    const data = google.visualization.arrayToDataTable([
    ['Time', 'TMP36 Sensor'],
    ['1',  22.5],
    ['2',  23.1],
    ['3',  24.4],
    ...
    ['9',  23.5],
    ['10',  23.9]
    ]);

    const options = {
    title: 'Temperature Data',
    hAxis: {title: 'Time [s]'},
    vAxis: {title: 'Temperature [°C]'},
    curveType: 'function',
    legend: { position: 'right' }
    };

    const chart = new
google.visualization.LineChart(document.getElementById('mychart'));
    chart.draw(data, options);
}
</script>

<body>
    <h1>Plotting Temperature Data</h1>
    <p>Here is the temperature data from the TMP36 sensor.</p>
    <div id="mychart" style="width: 1200px; height: 600px"></div>
</body>
</html>
```

Options

```
..

    const options = {
    title: 'Temperature Data',
    hAxis: {title: 'Time [s]'},
    vAxis: {title: 'Temperature [°C]'},
    curveType: 'function',
    legend: { position: 'right' }
    };


..
```

This is just some examples of the different options that you can use to customize your plot.

Resources

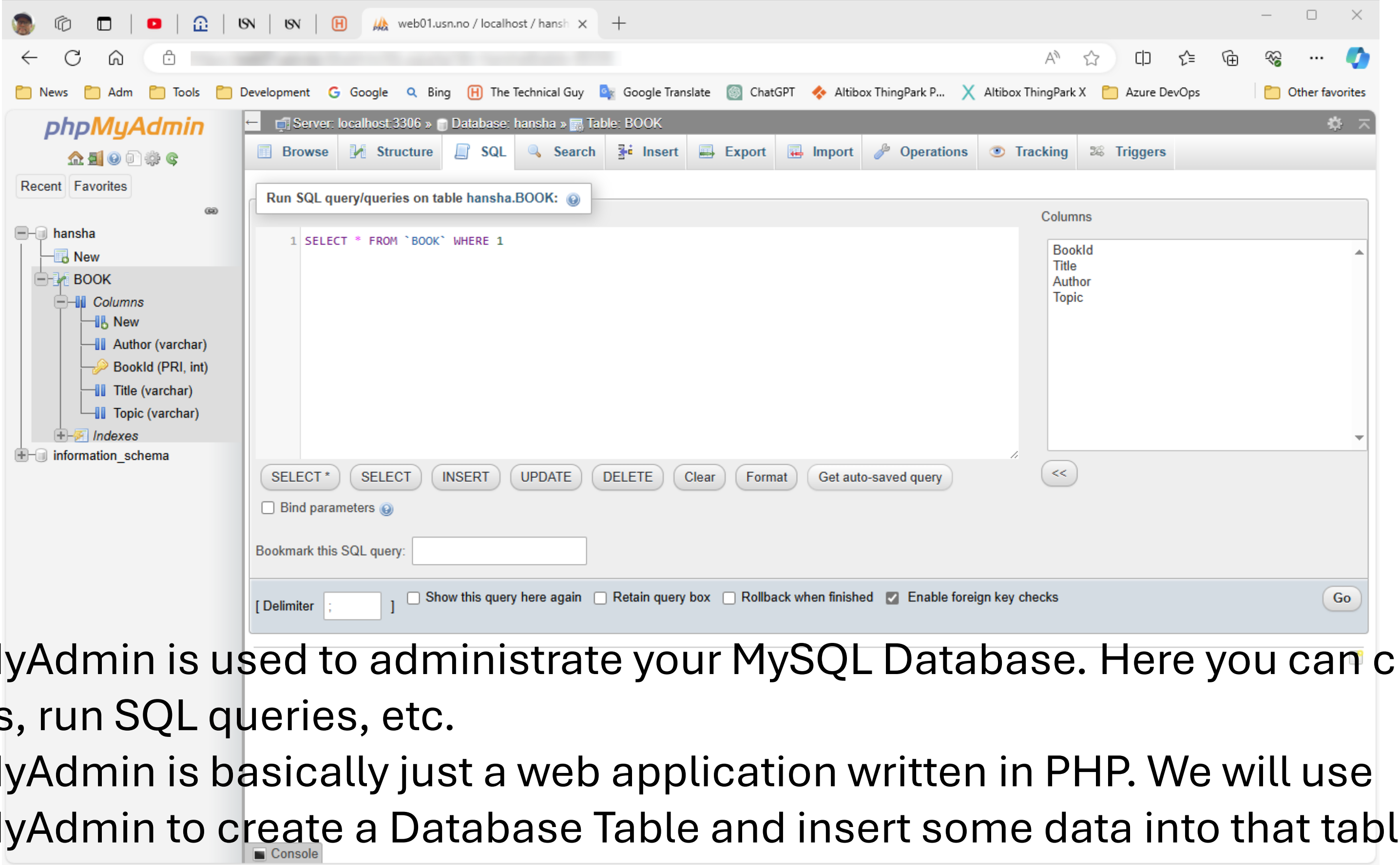
- Google Charts: https://developers.google.com/chart
- Google Chart Tutorial w3Schools: https://www.w3schools.com/js/js_graphics_google_chart.asp

Database

Table of Contents

Hans-Petter Halvorsen

phpMyAdmin



Server: localhost:3306 » Database: hansha » Table: BOOK

Run SQL query/queries on table hansha.BOOK:

```
SELECT * FROM `BOOK` WHERE 1
```

Columns: BookId, Title, Author, Topic

- hansha
  - New
  - BOOK
    - Columns
      - New
      - Author (varchar)
      - BookId (PRI, int)
      - Title (varchar)
      - Topic (varchar)
    - Indexes
- information_schema

□ Bind parameters

Bookmark this SQL query:

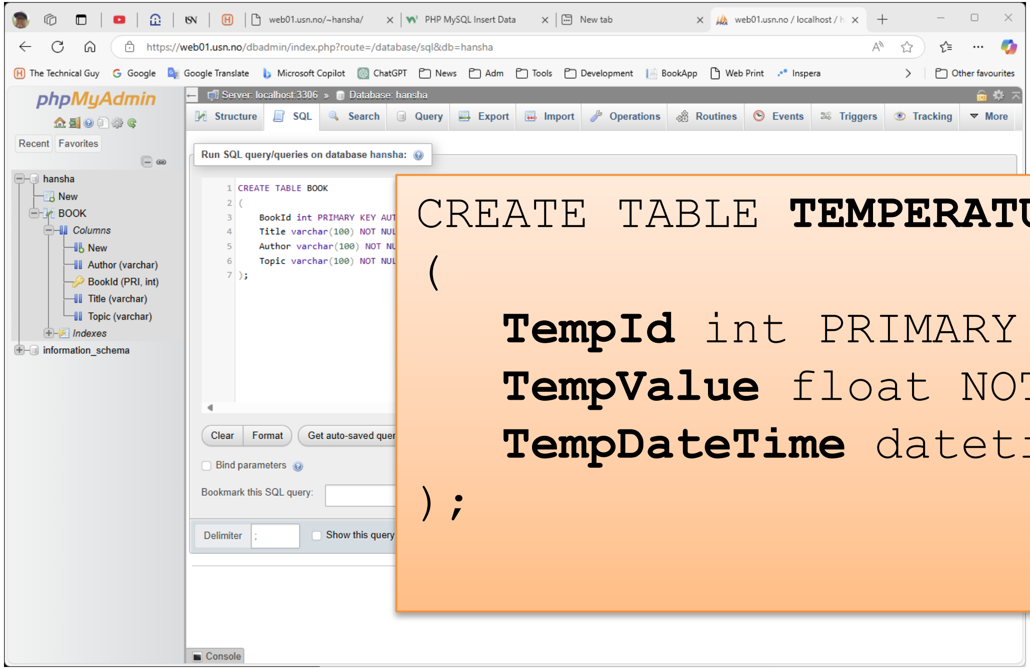
[Delimiter ;] □ Show this query here again □ Retain query box □ Rollback when finished ☑ Enable foreign key checks

phpMyAdmin is used to administrate your MySQL Database. Here you can create tables, run SQL queries, etc.

phpMyAdmin is basically just a web application written in PHP. We will use phpMyAdmin to create a Database Table and insert some data into that table.

Create Database

We can create Databases and Database Tables using PHP. But typically, we create a Database and the necessary Tables in advance before we start coding the Web Application. We use the phpMyAdmin tool.



https://web01.usn.no/dbadmin/index.php?route=/database/sql&db=hansha

Server: localhost:3306 » Database: hansha

Run SQL query/queries on database hansha:

```
CREATE TABLE BOOK
(
    BookId int PRIMARY KEY AUT
    Title varchar(100) NOT NUL
    Author varchar(100) NOT NU
    Topic varchar(100) NOT NUL
);
```

```
CREATE TABLE TEMPERATURE
(
    TempId int PRIMARY KEY AUTO_INCREMENT,
    TempValue float NOT NULL,
    TempDateTime datetime NOT NULL
);
```

Database

We can also insert some data into the Table using phpMyAdmin, e.g.:

```
insert into TEMPERATURE (TempValue, TempDateTime) values (22.5, ‘2025.02.05 12:00’);
insert into TEMPERATURE (TempValue, TempDateTime) values (23.1, ‘2025.02.05 12:10’);
insert into TEMPERATURE (TempValue, TempDateTime) values (22.3, ‘2025.02.05 12:20’);
insert into TEMPERATURE (TempValue, TempDateTime) values (24.3, ‘2025.02.05 12:30’);
insert into TEMPERATURE (TempValue, TempDateTime) values (25.4, ‘2025.02.05 12:40’);
insert into TEMPERATURE (TempValue, TempDateTime) values (21.3, ‘2025.02.05 12:50’);
insert into TEMPERATURE (TempValue, TempDateTime) values (22.3, ‘2025.02.05 13:00’);
insert into TEMPERATURE (TempValue, TempDateTime) values (23.4, ‘2025.02.05 13:10’);
insert into TEMPERATURE (TempValue, TempDateTime) values (24.3, ‘2025.02.05 13:20’);
insert into TEMPERATURE (TempValue, TempDateTime) values (23.3, ‘2025.02.05 13:30’);
```

PHP Server-side

Retrieving Data

Hans-Petter Halvorsen

Open Connection

In this tutorial we will use MySQLi. Here you see an example how we can connect to the database:

```
<?php
$servername = "localhost";
$dbname = "dbname";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully.";
?>
```

Close Connection after we have communicated with the database:

```
mysqli_close($conn);
```

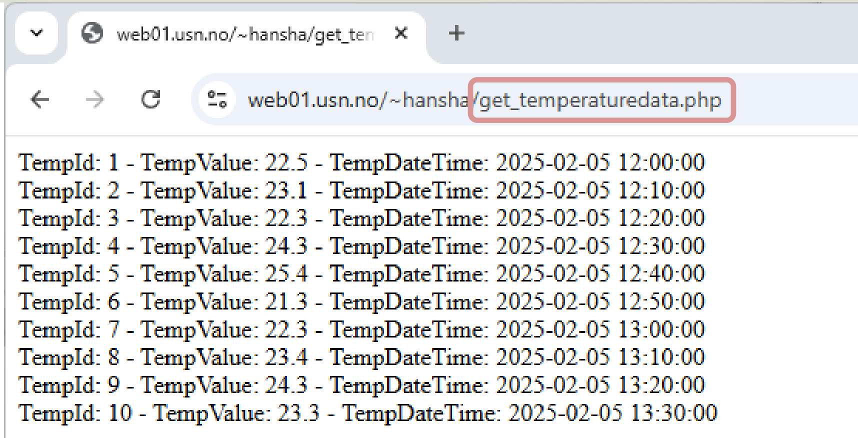
PHP Config File

Typically, we want to hide the Connection to the database, so, we can put it into a separate PHP file called, e.g., “config.php”. The in the different PHP files we can include this file. This file will contain username, password, etc. for the MySQL Server database.

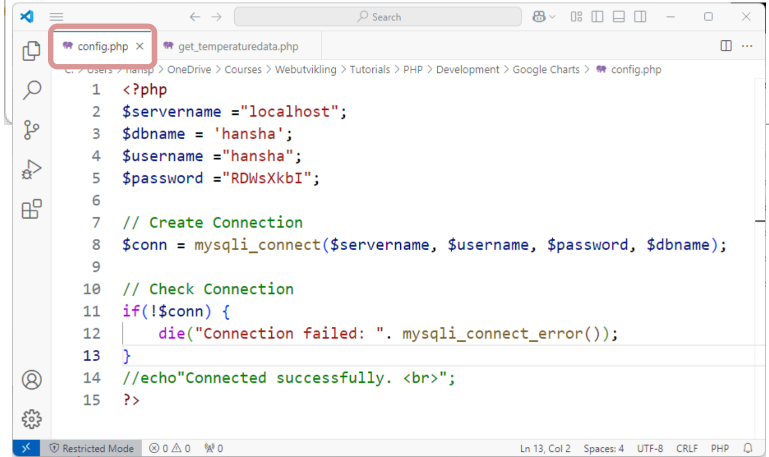
```
<?php
$servername ="localhost";
$username ="xxxxx";
$password ="xxxxx";
$dbname = "xxxxx";

// Create Connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check Connection
if(!$conn) {
    die("Connection failed: ". mysqli_connect_error());
}
echo"Connected successfully.";
?>
```

config.php

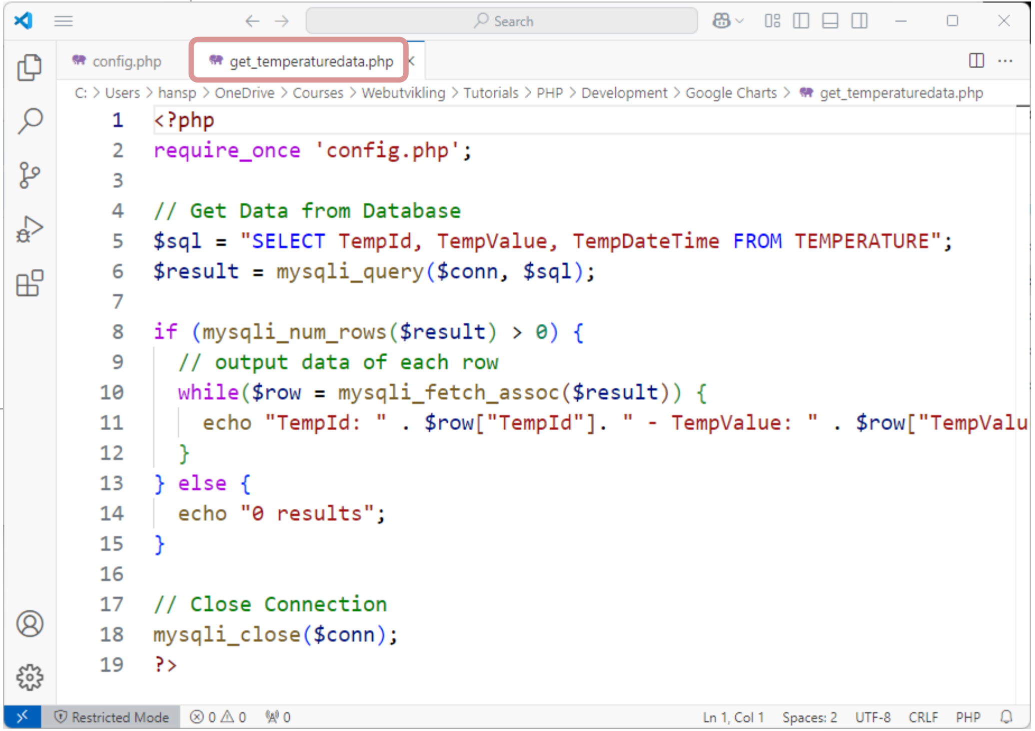
Show Data from the Database


web01.usn.no/~hansha/get_ten
web01.usn.no/~hansha/get_temperaturedata.php

```
TempId: 1 - TempValue: 22.5 - TempDateTime: 2025-02-05 12:00:00
TempId: 2 - TempValue: 23.1 - TempDateTime: 2025-02-05 12:10:00
TempId: 3 - TempValue: 22.3 - TempDateTime: 2025-02-05 12:20:00
TempId: 4 - TempValue: 24.3 - TempDateTime: 2025-02-05 12:30:00
TempId: 5 - TempValue: 25.4 - TempDateTime: 2025-02-05 12:40:00
TempId: 6 - TempValue: 21.3 - TempDateTime: 2025-02-05 12:50:00
TempId: 7 - TempValue: 22.3 - TempDateTime: 2025-02-05 13:00:00
TempId: 8 - TempValue: 23.4 - TempDateTime: 2025-02-05 13:10:00
TempId: 9 - TempValue: 24.3 - TempDateTime: 2025-02-05 13:20:00
TempId: 10 - TempValue: 23.3 - TempDateTime: 2025-02-05 13:30:00
```


config.php
get_temperaturedata.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Google Charts > config.php
Restricted Mode
Ln 13, Col 2   Spaces: 4   UTF-8   CRLF   PHP

```php
<?php
$servername ="localhost";
$dbname = 'hansha';
$username ="hansha";
$password ="RDWsXkbI";

// Create Connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check Connection
if(!$conn) {
    die("Connection failed: ". mysqli_connect_error());
}
//echo"Connected successfully. <br>";
?>
```


Search
config.php
get_temperaturedata.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Google Charts > get_temperaturedata.php
Restricted Mode
Ln 1, Col 1   Spaces: 2   UTF-8   CRLF   PHP

```php
<?php
require_once 'config.php';

// Get Data from Database
$sql = "SELECT TempId, TempValue, TempDateTime FROM TEMPERATURE";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
    echo "TempId: " . $row["TempId"]. " - TempValue: " . $row["TempValu
  }
} else {
  echo "0 results";
}

// Close Connection
mysqli_close($conn);
?>
```

https://www.halvorsen.blog

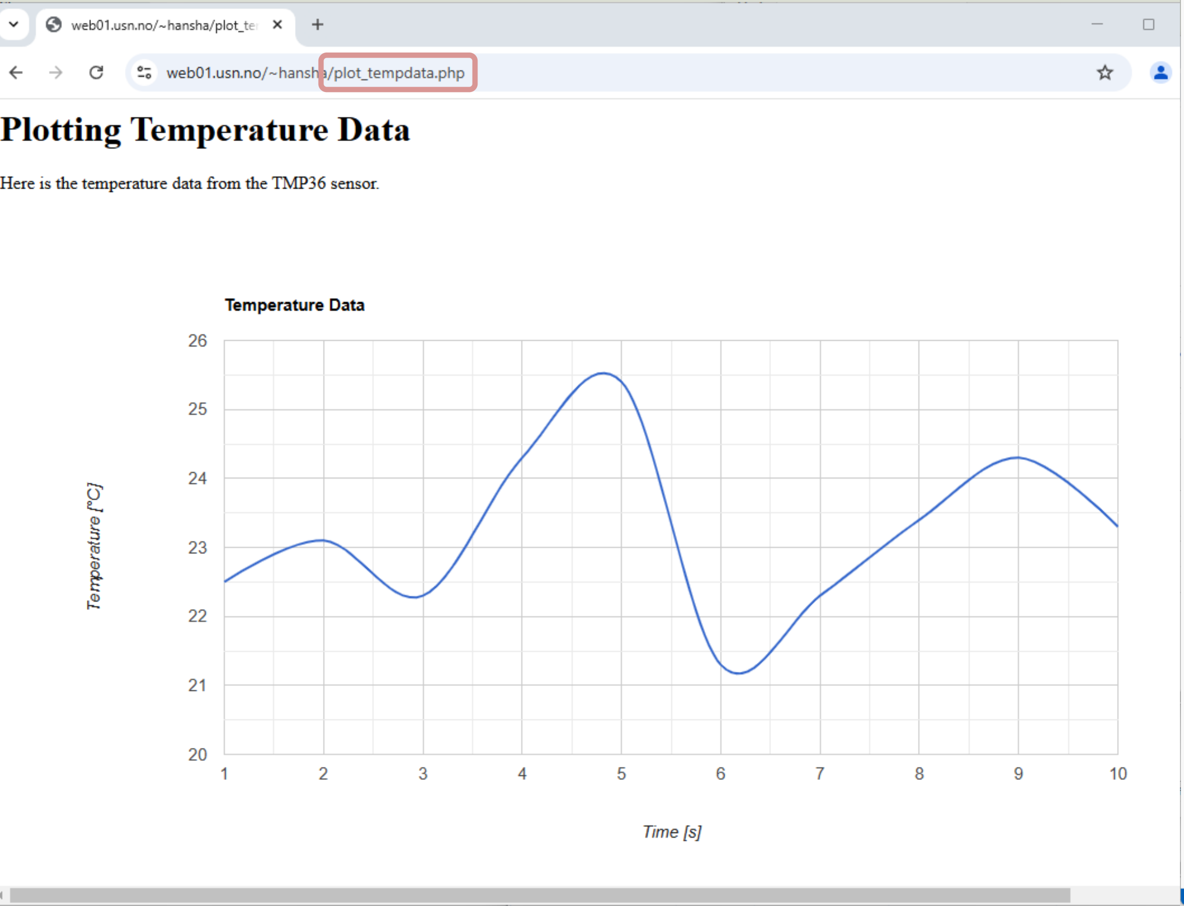
Plotting Data from Database

Using Google Charts on Client-side

Table of Contents

Hans-Petter Halvorsen

Plotting DB Data

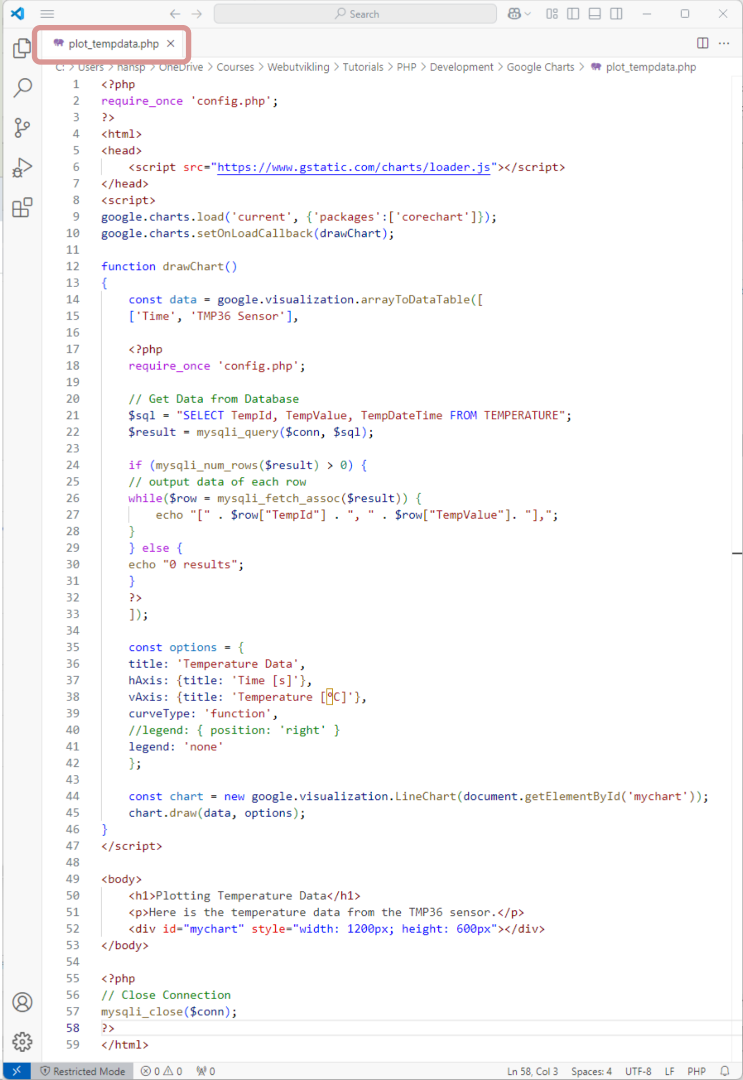


web01.usn.no/~hansha/plot_te
web01.usn.no/~hansha/plot_tempdata.php

Plotting Temperature Data

Here is the temperature data from the TMP36 sensor.

Temperature Data
26
25
24
23
22
21
20
Temperature [°C]
1
2
3
4
5
6
7
8
9
10
Time [s]



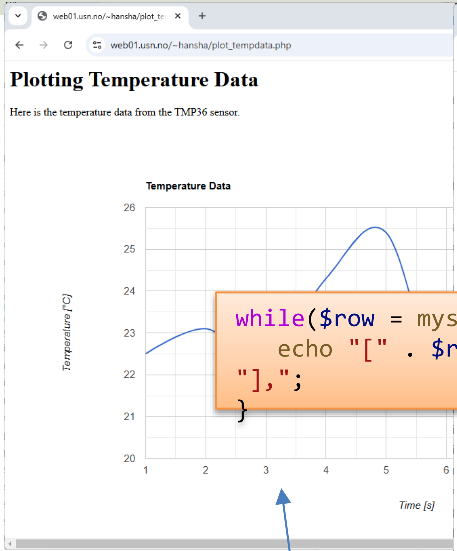
plot_tempdata.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Google Charts > plot_tempdata.php
Restricted Mode
Ln 58, Col 3
Spaces: 4
UTF-8
LF
PHP

```php
<?php
require_once 'config.php';
?>
<html>
<head>
    <script src="https://www.gstatic.com/charts/loader.js"></script>
</head>
<script>
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawChart);

function drawChart()
{
    const data = google.visualization.arrayToDataTable([
    ['Time', 'TMP36 Sensor'],

    <?php
    require_once 'config.php';

    // Get Data from Database
    $sql = "SELECT TempId, TempValue, TempDateTime FROM TEMPERATURE";
    $result = mysqli_query($conn, $sql);

    if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo "[" . $row["TempId"] . ", " . $row["TempValue"]. "],";
    }
    } else {
    echo "0 results";
    }
    ?>
    ]);

    const options = {
    title: 'Temperature Data',
    hAxis: {title: 'Time [s]'},
    vAxis: {title: 'Temperature [°C]'},
    curveType: 'function',
    //legend: { position: 'right' }
    legend: 'none'
    };

    const chart = new google.visualization.LineChart(document.getElementById('mychart'));
    chart.draw(data, options);
}
</script>

<body>
    <h1>Plotting Temperature Data</h1>
    <p>Here is the temperature data from the TMP36 sensor.</p>
    <div id="mychart" style="width: 1200px; height: 600px"></div>
</body>

<?php
// Close Connection
mysqli_close($conn);
?>
</html>
```

Code

```
function drawChart()
{
    const data = google.visualization.arrayToDataTable([
    ['Time', 'TMP36 Sensor'],

    <?php
    require_once 'config.php';

    // Get Data from Database
    $sql = "SELECT TempId, TempValue, TempDateTime FROM TEMPERATURE";
    $result = mysqli_query($conn, $sql);

    if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo "[" . $row["TempId"] . ", " . $row["TempValue"]. "],";
    }
    } else {
    echo "0 results";
    }
    ?>
    ]);

    const options = {
    title: 'Temperature Data',
    hAxis: {title: 'Time [s]'},
    vAxis: {title: 'Temperature [°C]'},
    curveType: 'function',
    //legend: { position: 'right' }
    legend: 'none'
    };

    const chart = new google.visualization.LineChart(document.getElementById('mychart'));
    chart.draw(data, options);
}
```

Different x-axis

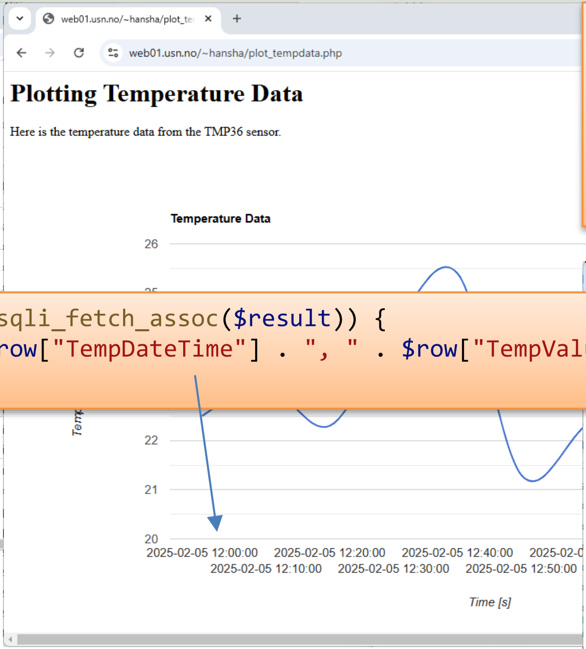


web01.usn.no/~hansha/plot_te
web01.usn.no/~hansha/plot_tempdata.php
Temperature Data
26
25
24
23
22
21
20
Temperature [°C]
1
2
3
4
5
6
Time [s]

Plotting Temperature Data

Here is the temperature data from the TMP36 sensor.

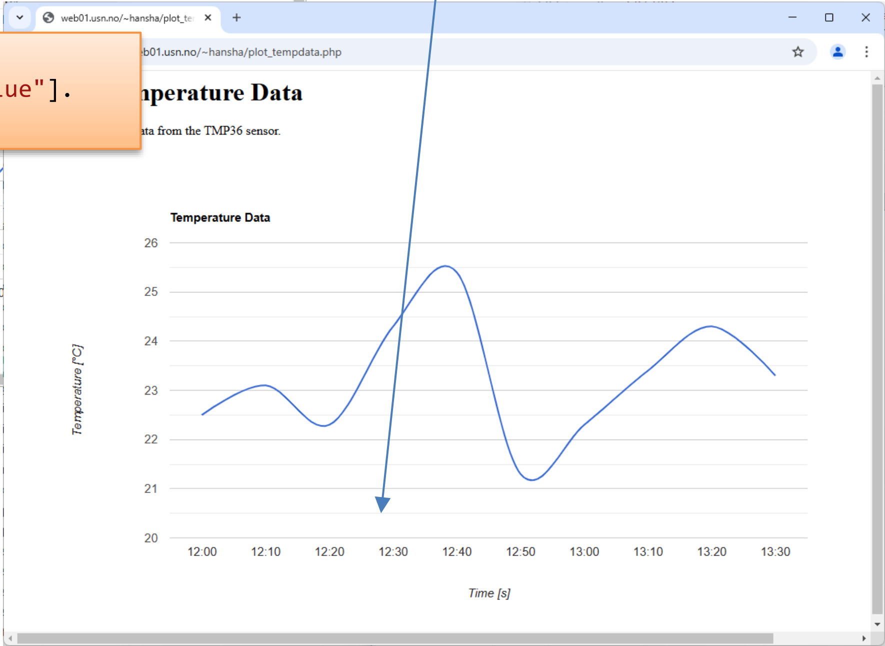
```
while($row = mysqli_fetch_assoc($result)) {
   echo "[" . $row["TempId"] . ", " . $row["TempValue"]. "],";
}
```



Plotting Temperature Data

Here is the temperature data from the TMP36 sensor.

```
while($row = mysqli_fetch_assoc($result)) {
   echo "[" . $row["TempDateTime"] . ", " . $row["TempValue"].
"],";
}
```



nperature Data

ata from the TMP36 sensor.

```
while($row = mysqli_fetch_assoc($result)) {
   $date = date_create($row["TempDateTime"]);
   $xaxis = date_format($date,"H:i");
   echo "['" . $xaxis . "', " . $row["TempValue"]. "],";
}
```

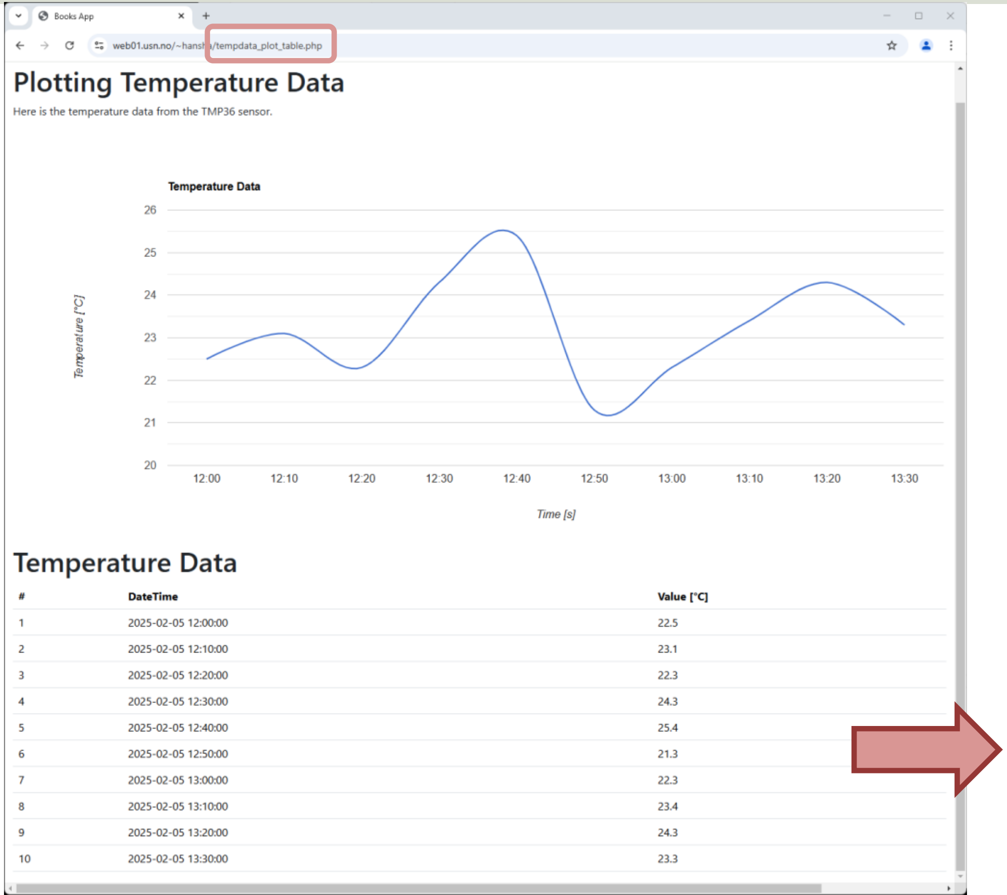
Improvements

Improvements

Here are some examples of improvements to make for this basic plotting application:

- Show Data in a Chart and, in addition, show Data in a HTML Table with Bootstrap for better visual appearance.
- Show Data from Multiple Temperature Sensors in the same chart.
- Select which Sensor to show Data from. Here we can use a “Dropdown” menu (i.e., use the HTML select tag).
- Select “From Date” and “To Date” to specify data to show in the Plot.
- In general, improve user interface, code structure and quality.

Plotting + Show Data in Table



Plotting Temperature Data

Here is the temperature data from the TMP36 sensor.

Temperature Data

#	DateTime	Value [°C]
1	2025-02-05 12:00:00	22.5
2	2025-02-05 12:10:00	23.1
3	2025-02-05 12:20:00	22.3
4	2025-02-05 12:30:00	24.3
5	2025-02-05 12:40:00	25.4
6	2025-02-05 12:50:00	21.3
7	2025-02-05 13:00:00	22.3
8	2025-02-05 13:10:00	23.4
9	2025-02-05 13:20:00	24.3
10	2025-02-05 13:30:00	23.3

```
<h1>Temperature Data</h1>
<div class="table-responsive">
<table class="table"">
  <thead>
    <tr>
      <th>#</th>
      <th>DateTime</th>
      <th>Value [°C]</th>
    </tr>
  </thead>

  <tbody>
  <?php
  // Get Data from Database
  $sql = "SELECT TempId, TempValue, TempDateTime FROM TEMPERATURE";
  $result = mysqli_query($conn, $sql);

  if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
      echo "<tr>";
      echo "<td>" . $row["TempId"] . "</td>";
      echo "<td>" . $row["TempDateTime"] . "</td>";
      echo "<td>" . $row["TempValue"] . "</td>";
      echo "</tr>";
    }
  } else {
    echo "0 results";
  }
  ?>

  </tbody>
</table>
</div>
```

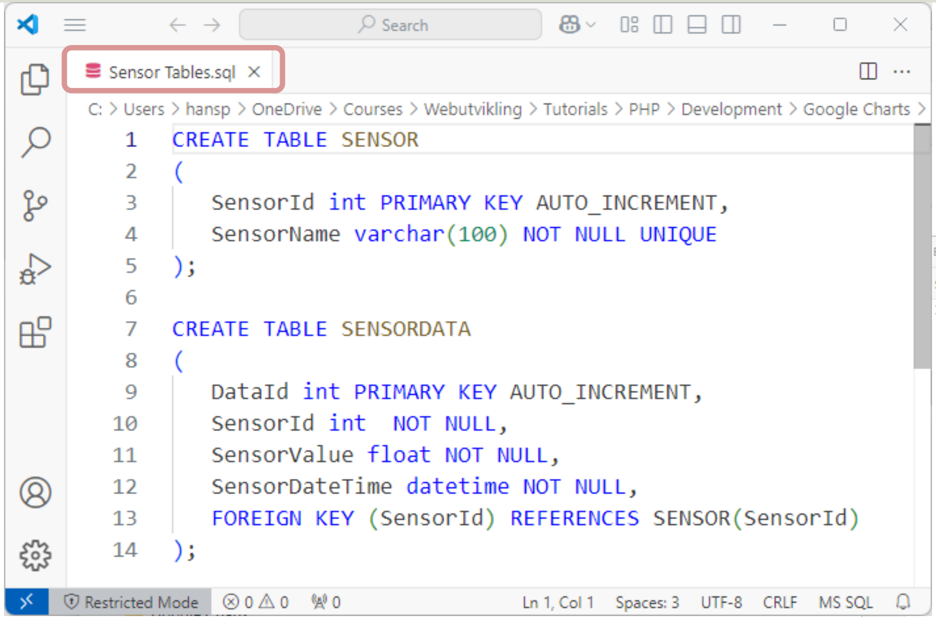
Plot Data from multiple Temperature Sensors?

So far, we have plotted data from one temperature sensor. What if we have multiple sensors? Here are some possible alternatives;

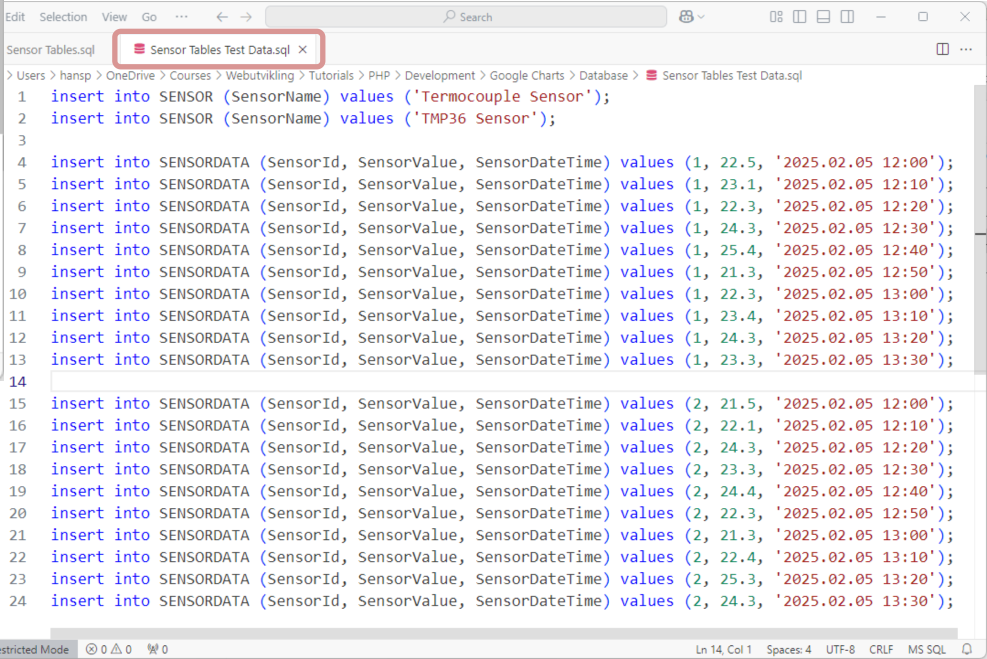
- Show Data from Multiple Temperature Sensors in the same Chart.
- Select which Sensor to show Data from. Here we can use a “Dropdown” menu (i.e., use the HTML select tag).

For both options we need to update our database structure to handle more than one temperature sensor.

Updated Database

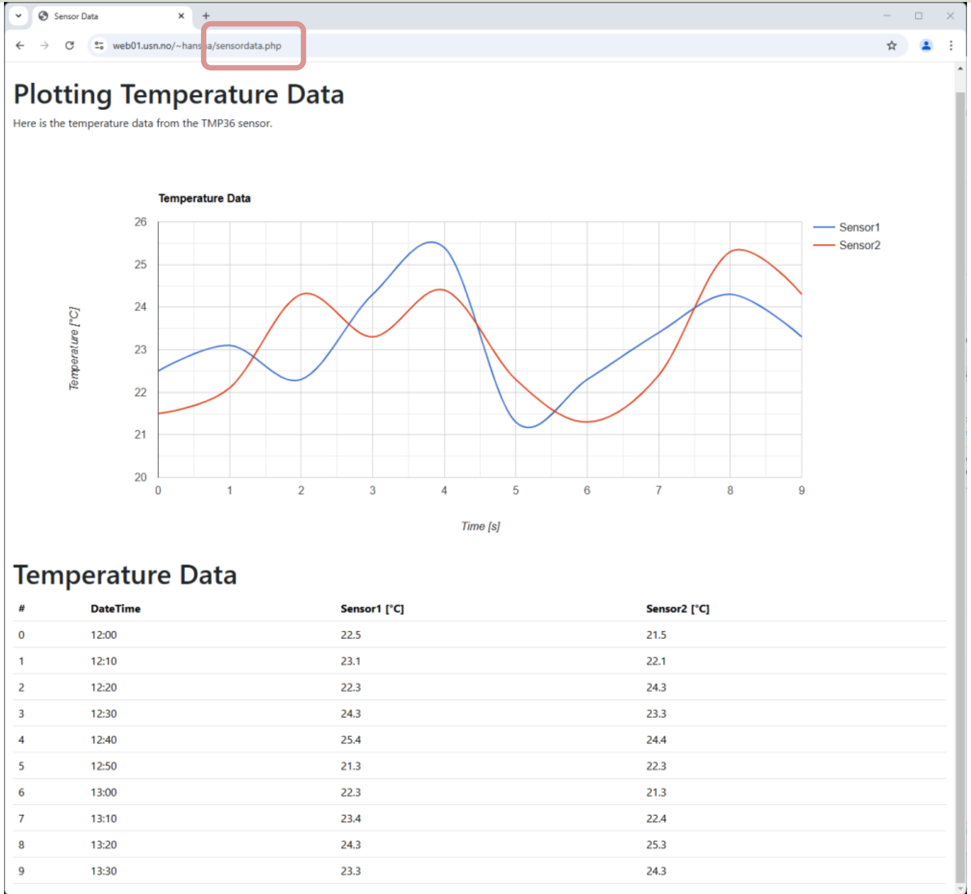

Search
Sensor Tables.sql
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Google Charts >
Restricted Mode
Ln 1, Col 1
Spaces: 3
UTF-8
CRLF
MS SQL

```sql
CREATE TABLE SENSOR
(
    SensorId int PRIMARY KEY AUTO_INCREMENT,
    SensorName varchar(100) NOT NULL UNIQUE
);

CREATE TABLE SENSORDATA
(
    DataId int PRIMARY KEY AUTO_INCREMENT,
    SensorId int  NOT NULL,
    SensorValue float NOT NULL,
    SensorDateTime datetime NOT NULL,
    FOREIGN KEY (SensorId) REFERENCES SENSOR(SensorId)
);
```


Edit
Selection
View
Go
Search
Sensor Tables.sql
Sensor Tables Test Data.sql
> Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Google Charts > Database > Sensor Tables Test Data.sql
Restricted Mode
Ln 14, Col 1
Spaces: 4
UTF-8
CRLF
MS SQL

```sql
insert into SENSOR (SensorName) values ('Termocouple Sensor');
insert into SENSOR (SensorName) values ('TMP36 Sensor');

insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (1, 22.5, '2025.02.05 12:00');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (1, 23.1, '2025.02.05 12:10');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (1, 22.3, '2025.02.05 12:20');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (1, 24.3, '2025.02.05 12:30');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (1, 25.4, '2025.02.05 12:40');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (1, 21.3, '2025.02.05 12:50');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (1, 22.3, '2025.02.05 13:00');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (1, 23.4, '2025.02.05 13:10');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (1, 24.3, '2025.02.05 13:20');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (1, 23.3, '2025.02.05 13:30');

insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (2, 21.5, '2025.02.05 12:00');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (2, 22.1, '2025.02.05 12:10');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (2, 24.3, '2025.02.05 12:20');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (2, 23.3, '2025.02.05 12:30');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (2, 24.4, '2025.02.05 12:40');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (2, 22.3, '2025.02.05 12:50');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (2, 21.3, '2025.02.05 13:00');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (2, 22.4, '2025.02.05 13:10');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (2, 25.3, '2025.02.05 13:20');
insert into SENSORDATA (SensorId, SensorValue, SensorDateTime) values (2, 24.3, '2025.02.05 13:30');
```

Plot Data from multiple Temperature Sensors



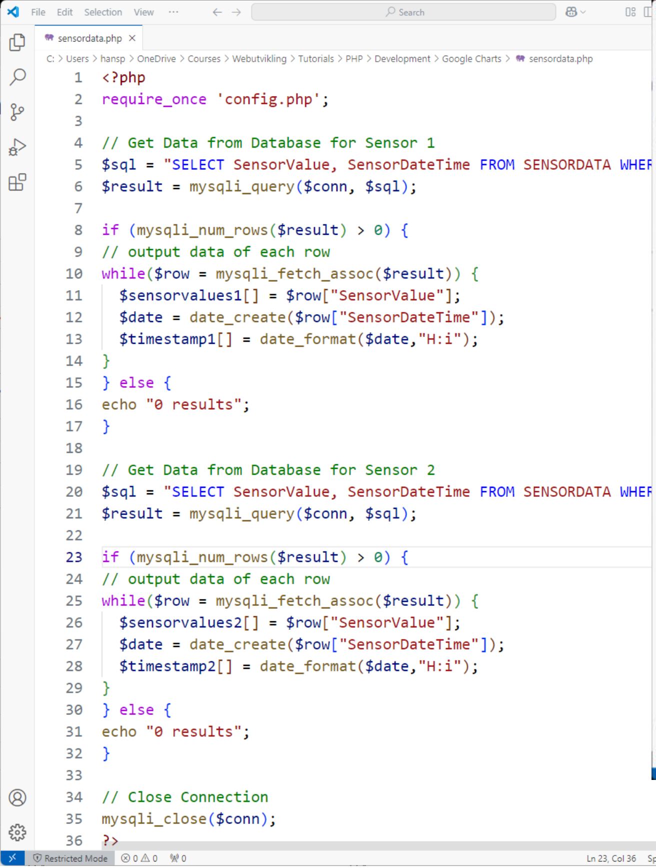
Plotting Temperature Data

Here is the temperature data from the TMP36 sensor.

Temperature Data

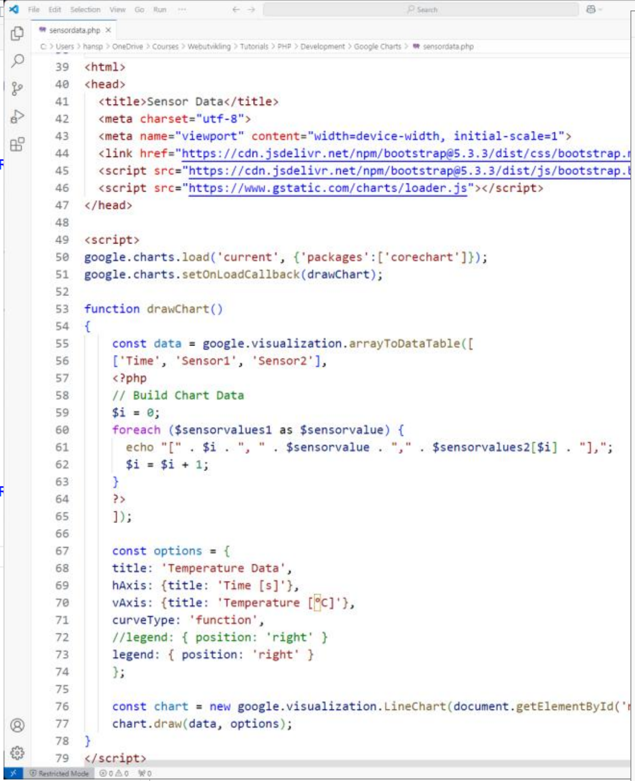
#	DateTime	Sensor1 [°C]	Sensor2 [°C]
0	12:00	22.5	21.5
1	12:10	23.1	22.1
2	12:20	22.3	24.3
3	12:30	24.3	23.3
4	12:40	25.4	24.4
5	12:50	21.3	22.3
6	13:00	22.3	21.3
7	13:10	23.4	22.4
8	13:20	24.3	25.3
9	13:30	23.3	24.3

Here is the updated example where we show data from 2 different sensors in the same plot.



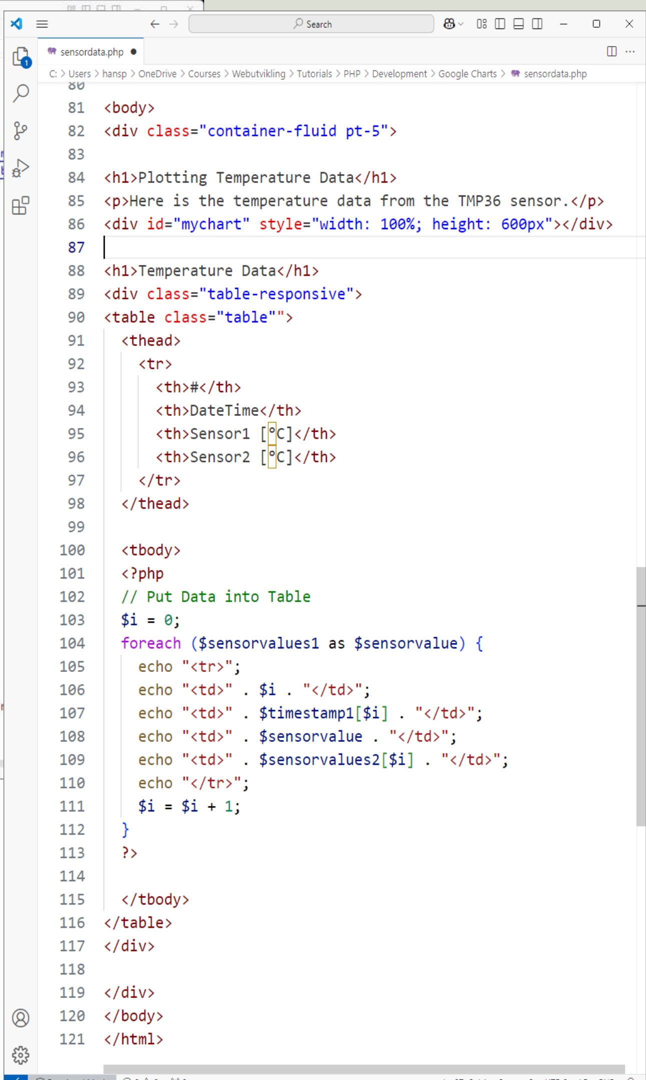
File Edit Selection View ···
Search
sensordata.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Google Charts > sensordata.php
Restricted Mode
Ln 23, Col 36 Spaces: 2 UTF-8 LF PHP

```
<?php
require_once 'config.php';

// Get Data from Database for Sensor 1
$sql = "SELECT SensorValue, SensorDateTime FROM SENSORDATA WHEF
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
// output data of each row
while($row = mysqli_fetch_assoc($result)) {
  $sensorvalues1[] = $row["SensorValue"];
  $date = date_create($row["SensorDateTime"]);
  $timestamp1[] = date_format($date,"H:i");
}
} else {
echo "0 results";
}

// Get Data from Database for Sensor 2
$sql = "SELECT SensorValue, SensorDateTime FROM SENSORDATA WHEF
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
// output data of each row
while($row = mysqli_fetch_assoc($result)) {
  $sensorvalues2[] = $row["SensorValue"];
  $date = date_create($row["SensorDateTime"]);
  $timestamp2[] = date_format($date,"H:i");
}
} else {
echo "0 results";
}

// Close Connection
mysqli_close($conn);
?>
```



File Edit Selection View Go Run ···
Search
sensordata.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Google Charts > sensordata.php
Restricted Mode

```
<html>
<head>
  <title>Sensor Data</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.r
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.b
  <script src="https://www.gstatic.com/charts/loader.js"></script>
</head>

<script>
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawChart);

function drawChart()
{
    const data = google.visualization.arrayToDataTable([
    ['Time', 'Sensor1', 'Sensor2'],
    <?php
    // Build Chart Data
    $i = 0;
    foreach ($sensorvalues1 as $sensorvalue) {
      echo "[" . $i . ", " . $sensorvalue . "," . $sensorvalues2[$i] . "],";
      $i = $i + 1;
    }
    ?>
    ]);

    const options = {
    title: 'Temperature Data',
    hAxis: {title: 'Time [s]'},
    vAxis: {title: 'Temperature [°C]'},
    curveType: 'function',
    //legend: { position: 'right' }
    legend: { position: 'right' }
    };

    const chart = new google.visualization.LineChart(document.getElementById('r
    chart.draw(data, options);
}
</script>
```

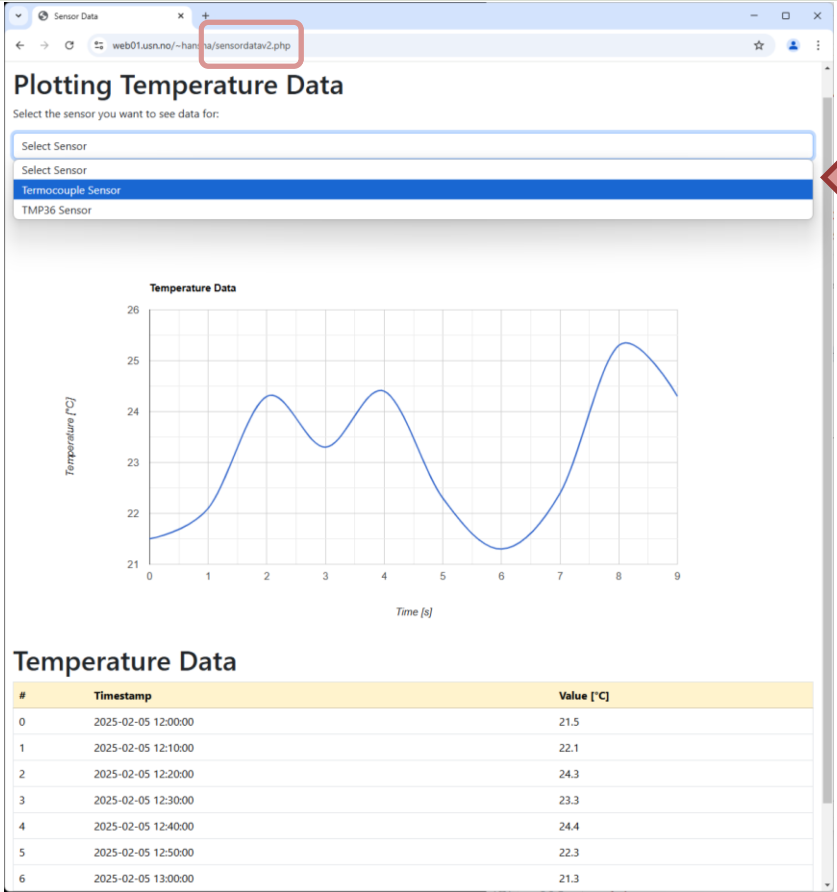


Search
sensordata.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Google Charts > sensordata.php

```
<body>
<div class="container-fluid pt-5">

<h1>Plotting Temperature Data</h1>
<p>Here is the temperature data from the TMP36 sensor.</p>
<div id="mychart" style="width: 100%; height: 600px"></div>

<h1>Temperature Data</h1>
<div class="table-responsive">
<table class="table"">
  <thead>
    <tr>
      <th>#</th>
      <th>DateTime</th>
      <th>Sensor1 [°C]</th>
      <th>Sensor2 [°C]</th>
    </tr>
  </thead>

  <tbody>
  <?php
  // Put Data into Table
  $i = 0;
  foreach ($sensorvalues1 as $sensorvalue) {
    echo "<tr>";
    echo "<td>" . $i . "</td>";
    echo "<td>" . $timestamp1[$i] . "</td>";
    echo "<td>" . $sensorvalue . "</td>";
    echo "<td>" . $sensorvalues2[$i] . "</td>";
    echo "</tr>";
    $i = $i + 1;
  }
  ?>

  </tbody>
</table>
</div>

</div>
</body>
</html>
```

Here is the updated code to make it possible to show data from 2 different sensors in the same plot. It works, but both SQL queries and PHP can be further improved.

Plot Data from multiple Temperature Sensors



Plotting Temperature Data

Select the sensor you want to see data for:

Select Sensor

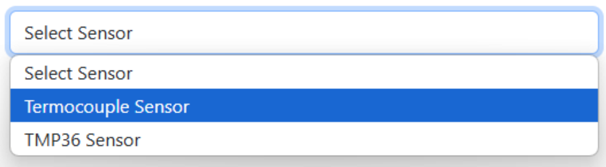
- Select Sensor
- Termocouple Sensor
- TMP36 Sensor

Temperature Data

#	Timestamp	Value [°C]
0	2025-02-05 12:00:00	21.5
1	2025-02-05 12:10:00	22.1
2	2025-02-05 12:20:00	24.3
3	2025-02-05 12:30:00	23.3
4	2025-02-05 12:40:00	24.4
5	2025-02-05 12:50:00	22.3
6	2025-02-05 13:00:00	21.3

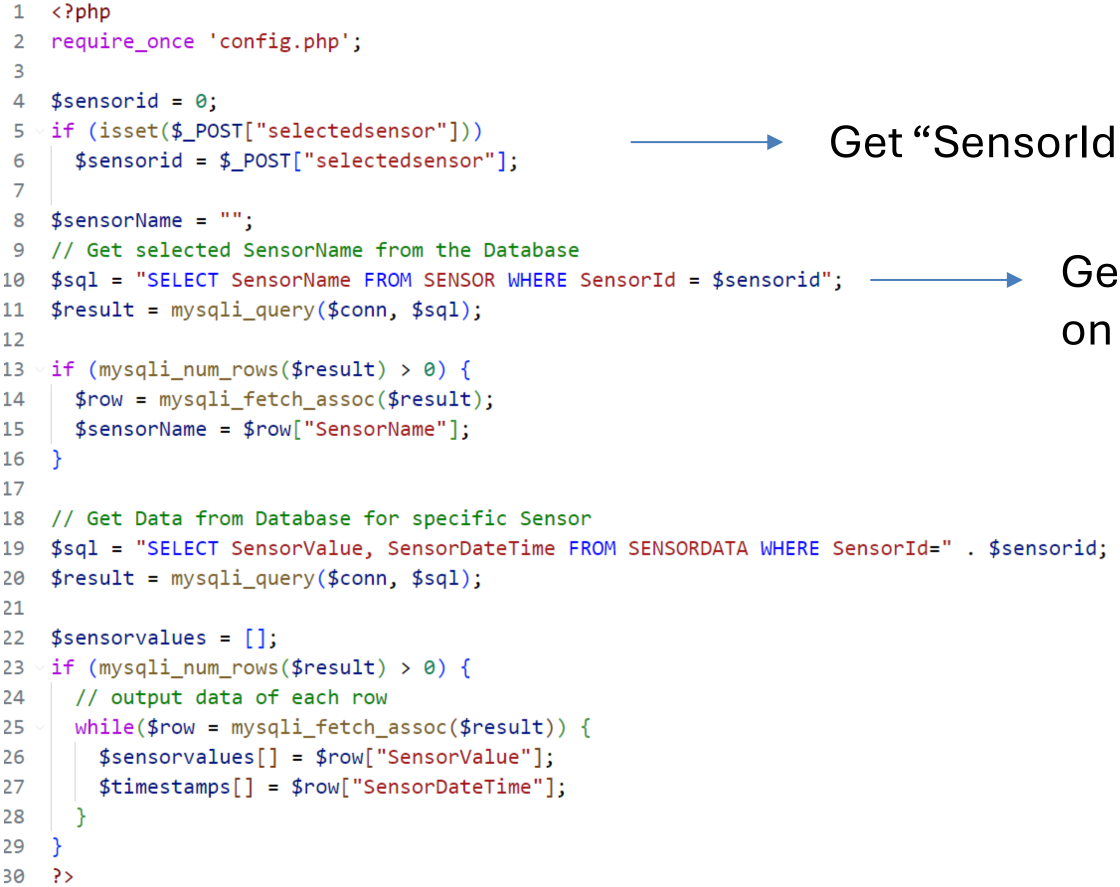
Here is another example where we need to select a specific sensor and then the data for that specific sensor is shown in the plot and in the table.

HTML Select



```
<form action="" method="POST">

<p>Select the sensor you want to see data for:</p>

<select name="selectedsensor" id="selectedsensor" class="form-control" onChange="this.form.submit()">
  <option value="0">Select Sensor</option>
  <?php
  // Get list of Sensors from the Database
  $sql = "SELECT SensorId, SensorName FROM SENSOR ORDER BY SensorName";
  $result = mysqli_query($conn, $sql);

  if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
      echo "<option value='" . $row["SensorId"] . "'>" . $row["SensorName"] . "</option>";
    }
  }
  ?>
</select><br><br>

</form>
```

Retrieve Data for a specific Sensor



```php
<?php
require_once 'config.php';

$sensorid = 0;
if (isset($_POST["selectedsensor"]))
  $sensorid = $_POST["selectedsensor"];

$sensorName = "";
// Get selected SensorName from the Database
$sql = "SELECT SensorName FROM SENSOR WHERE SensorId = $sensorid";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  $row = mysqli_fetch_assoc($result);
  $sensorName = $row["SensorName"];
}

// Get Data from Database for specific Sensor
$sql = "SELECT SensorValue, SensorDateTime FROM SENSORDATA WHERE SensorId=" . $sensorid;
$result = mysqli_query($conn, $sql);

$sensorvalues = [];
if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
    $sensorvalues[] = $row["SensorValue"];
    $timestamps[] = $row["SensorDateTime"];
  }
}
?>
```

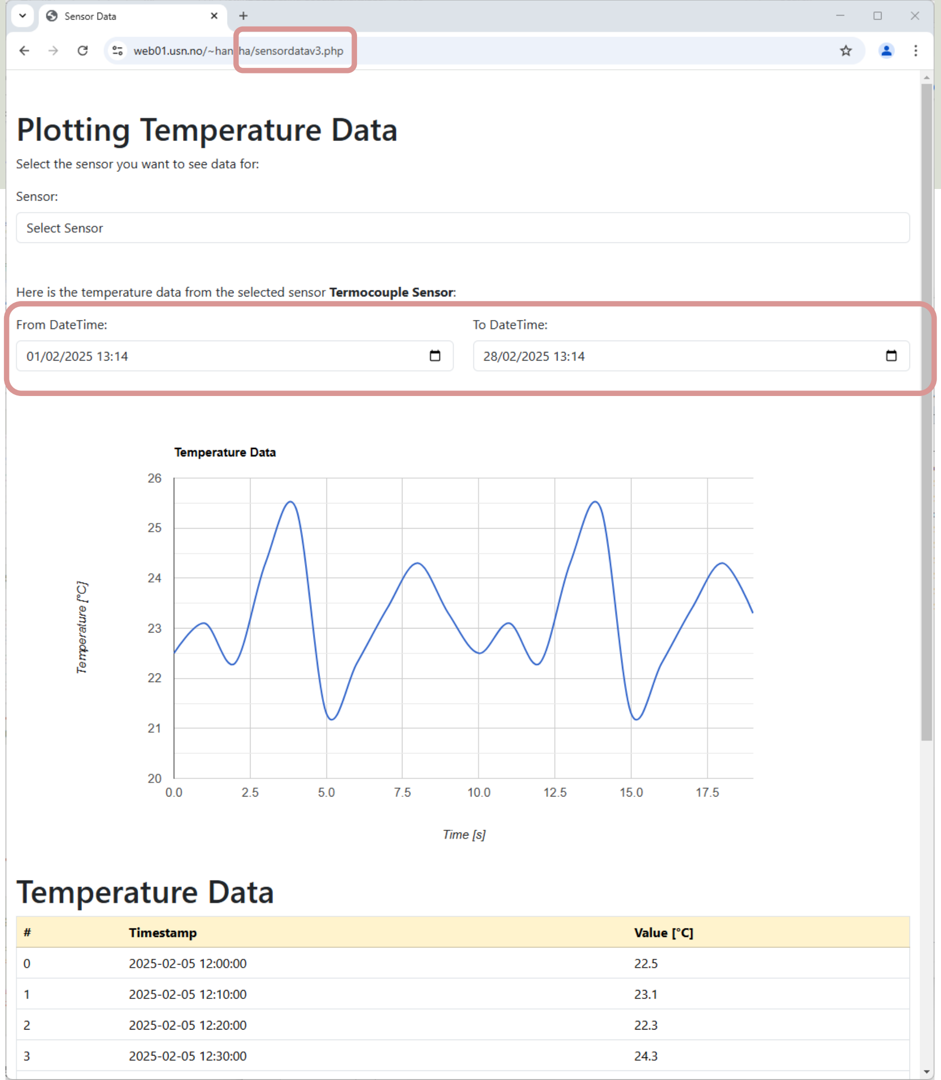
Get “SensorId” from postback

Get “SensorName” based on selected “SensorId”

Get Sensor Data based on selected “SensorId”

From/To Date

Here we have added “From DateTime” and “To DateTime” to make it possible to limit the data and show data only for a specific interval. This is important if we ,e.g., have data for many years.



Plotting Temperature Data

Select the sensor you want to see data for:

Sensor:

Select Sensor

Here is the temperature data from the selected sensor **Termocouple Sensor**:

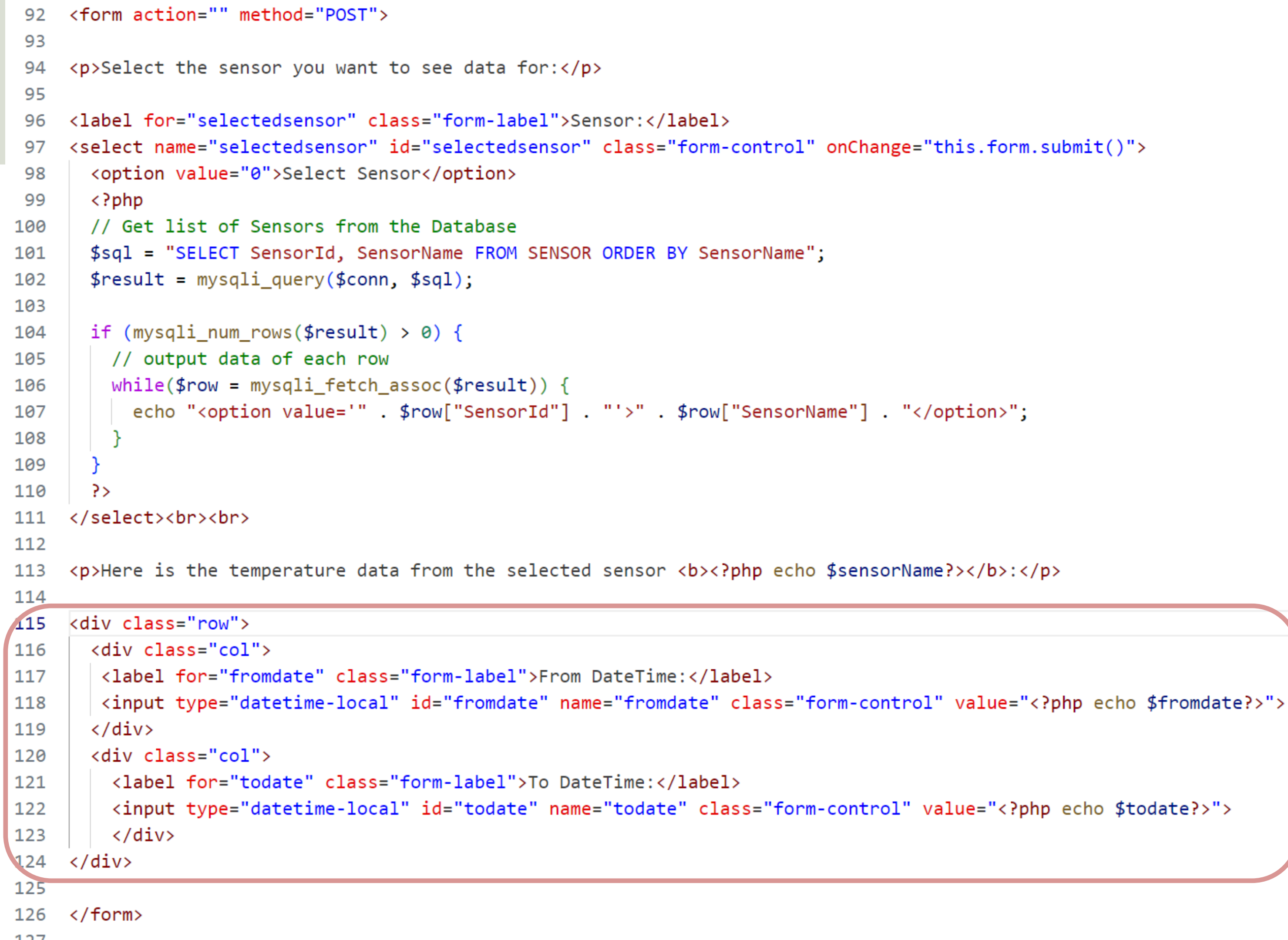
From DateTime: 01/02/2025 13:14

To DateTime: 28/02/2025 13:14

Temperature Data

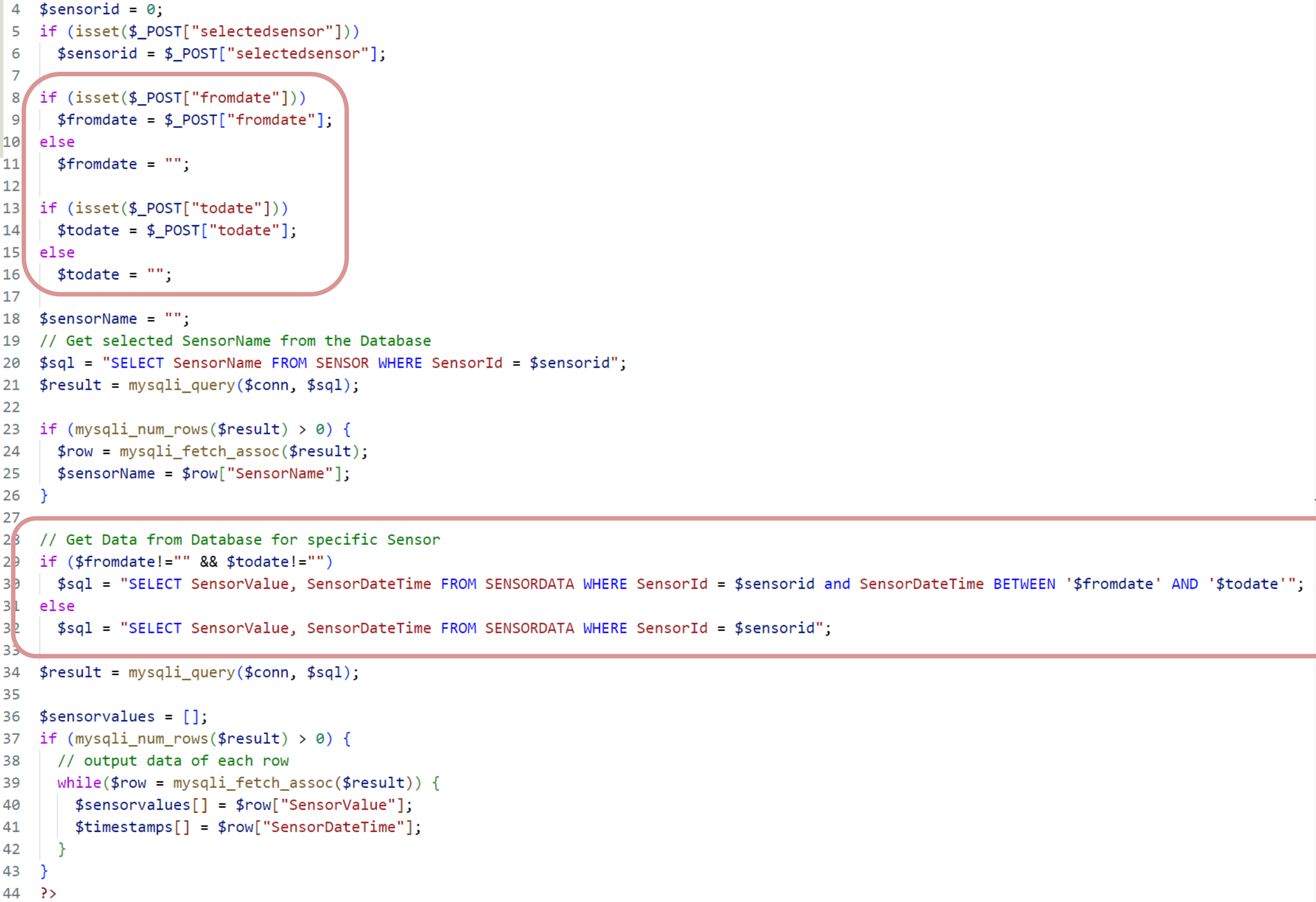
#	Timestamp	Value [°C]
0	2025-02-05 12:00:00	22.5
1	2025-02-05 12:10:00	23.1
2	2025-02-05 12:20:00	22.3
3	2025-02-05 12:30:00	24.3

Code



```
<form action="" method="POST">

<p>Select the sensor you want to see data for:</p>

<label for="selectedsensor" class="form-label">Sensor:</label>
<select name="selectedsensor" id="selectedsensor" class="form-control" onChange="this.form.submit()">
  <option value="0">Select Sensor</option>
  <?php
  // Get list of Sensors from the Database
  $sql = "SELECT SensorId, SensorName FROM SENSOR ORDER BY SensorName";
  $result = mysqli_query($conn, $sql);

  if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
      echo "<option value='" . $row["SensorId"] . "'>" . $row["SensorName"] . "</option>";
    }
  }
  ?>
</select><br><br>

<p>Here is the temperature data from the selected sensor <b><?php echo $sensorName?></b>:</p>

<div class="row">
  <div class="col">
   <label for="fromdate" class="form-label">From DateTime:</label>
   <input type="datetime-local" id="fromdate" name="fromdate" class="form-control" value="<?php echo $fromdate?>">
  </div>
  <div class="col">
    <label for="todate" class="form-label">To DateTime:</label>
    <input type="datetime-local" id="todate" name="todate" class="form-control" value="<?php echo $todate?>">
    </div>
</div>

</form>
```

Code



```
$sensorid = 0;
if (isset($_POST["selectedsensor"]))
  $sensorid = $_POST["selectedsensor"];

if (isset($_POST["fromdate"]))
  $fromdate = $_POST["fromdate"];
else
  $fromdate = "";

if (isset($_POST["todate"]))
  $todate = $_POST["todate"];
else
  $todate = "";

$sensorName = "";
// Get selected SensorName from the Database
$sql = "SELECT SensorName FROM SENSOR WHERE SensorId = $sensorid";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  $row = mysqli_fetch_assoc($result);
  $sensorName = $row["SensorName"];
}

// Get Data from Database for specific Sensor
if ($fromdate!="" && $todate!="")
  $sql = "SELECT SensorValue, SensorDateTime FROM SENSORDATA WHERE SensorId = $sensorid and SensorDateTime BETWEEN '$fromdate' AND '$todate'";
else
  $sql = "SELECT SensorValue, SensorDateTime FROM SENSORDATA WHERE SensorId = $sensorid";

$result = mysqli_query($conn, $sql);

$sensorvalues = [];
if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
    $sensorvalues[] = $row["SensorValue"];
    $timestamps[] = $row["SensorDateTime"];
  }
}
?>
```

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog