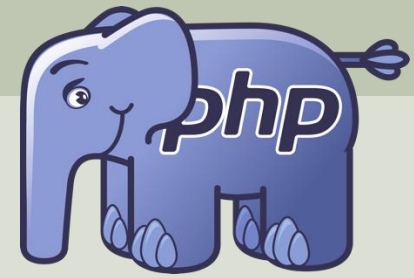


<https://www.halvorsen.blog>



PHP Tutorial

Hans-Petter Halvorsen



Contents

- [Introduction to Web Development](#)
- [Introduction to PHP](#)
- [Getting Started with PHP Programming](#)
- [PHP Programming](#)
- [HTML Forms in PHP](#)
- [Session variables](#)
- [PHP and MySQL Database](#)

Expected knowledge: You should already know basic HTML and have basic programming skills.

<https://www.halvorsen.blog>

Introduction



[Table of Contents](#)

Hans-Petter Halvorsen

Web Development Frameworks

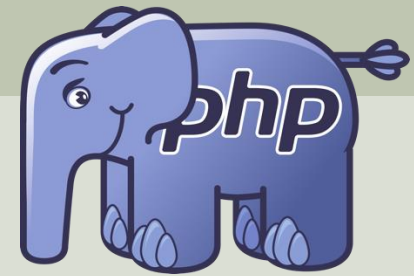
Some of the most used server-side (backend) Web Development Frameworks:

- **PHP** (Scripting language for Web development)
- **ASP.NET** (Microsoft, Visual Studio, C#)
- **Ruby on Rails** (Ruby)
- **Django** (Python)

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.
- 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).

<https://www.halvorsen.blog>



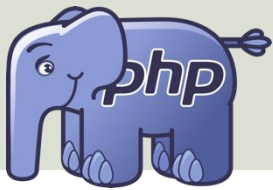
PHP

Hans-Petter Halvorsen



PHP

- PHP is a **server scripting language** for making dynamic and interactive web pages.
- PHP scripts are executed on the server/webserver.
- PHP files have extension ".php" and are typically a mix of PHP, HTML, CSS and JavaScript
- PHP is free and open-source.
- With PHP you can easily communicate with a Database, and especially MySQL.
- LAMP: Linux, Apache, MySQL and PHP.
- PHP is widely used and still by far the most used/popular language for web development.
- PHP is easy to learn (but still very powerful) – which cannot be said on many other web technologies and programming languages.
- Homepage: <https://www.php.net>
- PHP Tutorial: <https://www.w3schools.com/php>



PHP + MySQL



- You need to have a **PHP** + **MySQL** Environment on your local computer or get access to it from a server/Internet.
- For local installation you need to download and install Apache, PHP and MySQL.
- You can get server access from many providers (free or paid).
- (I will use an internal **LAMP** server available for employees and students at my University.)

LAMP

- LAMP = **L**inux, **A**pache, **M**ySQL, **P**HP
 - PHP is the Programming Language
 - MySQL is the Database System
 - Apache is the Web Server software
 - Linux is the operating system where the Web Server is running

Each part in LAMP is free and open-source, so it is a popular web hosting environment. You find also lots of online documentation and a large community.

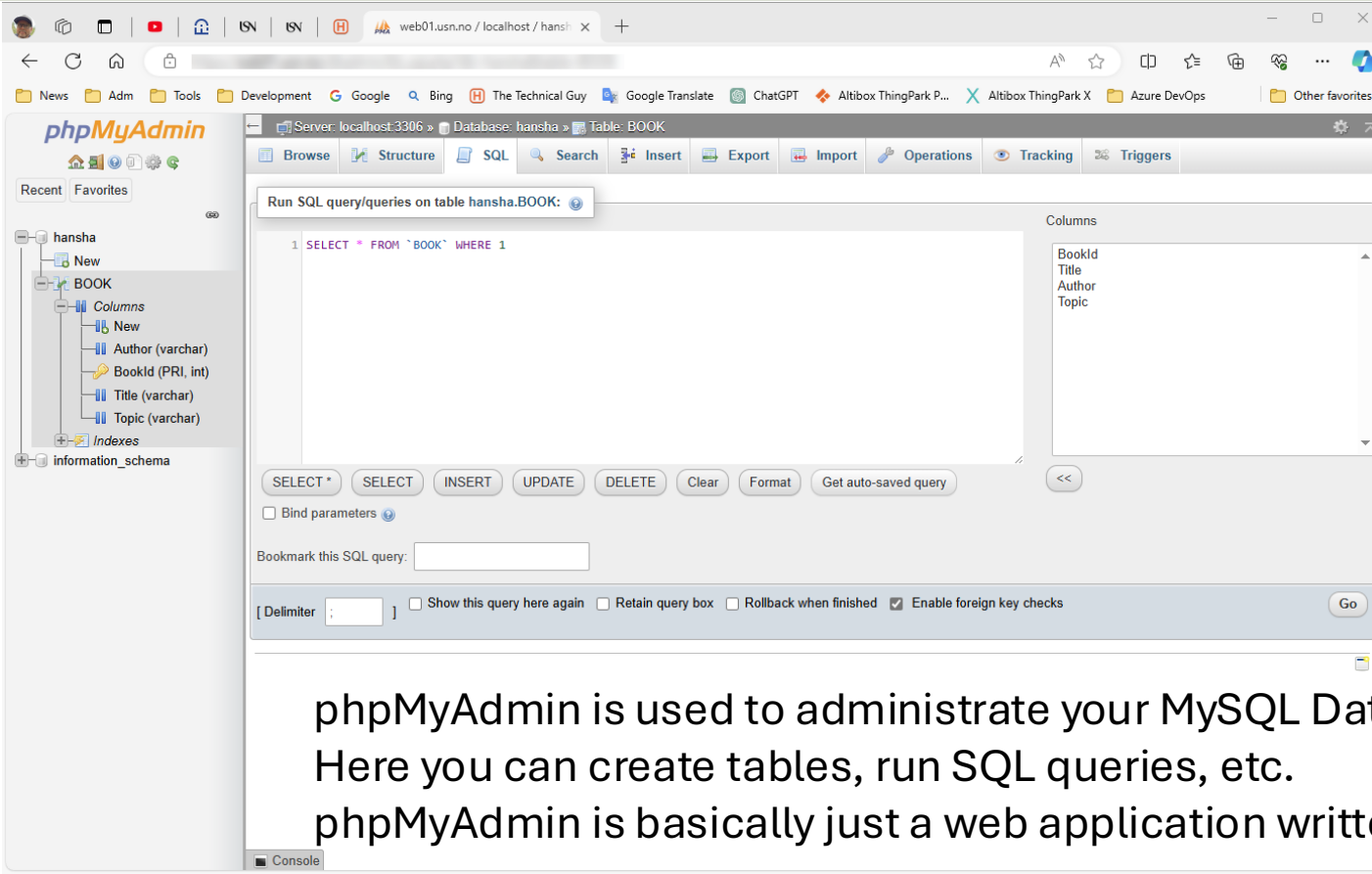
LAMP/PHP Web Hosting

- There exists hundreds/thousands of different LAMP/PHP/MySQL Hosting Providers, some free but mostly paid options.
- Hostinger - <https://www.hostinger.no>
- InfinityFree - <https://www.infinityfree.com>
- PRO ISP - <https://www.proisp.no>
- +++ (Just Google)

XAMPP

- XAMPP is a popular PHP development environment
- It works on Windows, macOS and Linux
- It installs Apache, MariaDB and PHP
 - Apache is a Web Server
 - MariaDB is almost identical to MySQL
 - PHP is the Web Programming language
- <https://www.apachefriends.org>

phpMyAdmin



The screenshot displays the phpMyAdmin web interface in a browser window. The address bar shows the URL `web01.usn.no / localhost / hansha`. The interface includes a navigation menu on the left with a tree view showing the database structure: `hansha` (parent), `New` (child), `BOOK` (child), `Columns` (child of `BOOK`), `Indexes` (child of `BOOK`), and `information_schema` (parent). The main content area is titled "Run SQL query/queries on table hansha.BOOK:" and contains a text input field with the SQL query `1 SELECT * FROM `BOOK` WHERE 1`. Below the query field are buttons for `SELECT *`, `SELECT`, `INSERT`, `UPDATE`, `DELETE`, `Clear`, `Format`, and `Get auto-saved query`. There is also a checkbox for `Bind parameters`. At the bottom, there is a "Bookmark this SQL query:" field, a "Delimiter" dropdown set to `:`, and several checkboxes: `Show this query here again`, `Retain query box`, `Rollback when finished`, and `Enable foreign key checks` (checked). A `Go` button is located at the bottom right of the query editor area.

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.

PHP Code Editors

You can use any kind of tool/IDE for Python(Django development). Here are some recommendations:

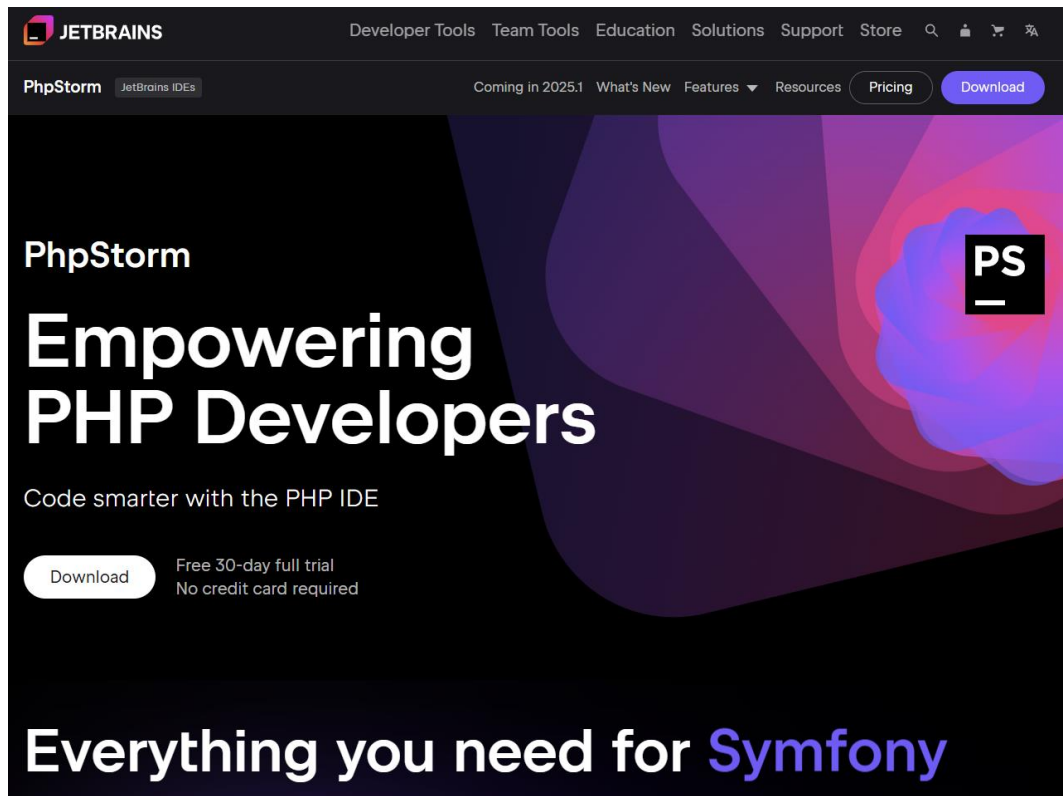
- Visual Studio Code.
 - Multiplatform and Free.
 - Homepage: <https://code.visualstudio.com>
- PhpStorm.
 - Free 30-day full trial.
 - Free access to all JetBrains IDEs for students and educators.
 - Homepage: <https://www.jetbrains.com>
- +++

Visual Studio Code

- Visual Studio Code is a free and open-source code editor developed by Microsoft.
- It supports many programming languages, including PHP, through an extension.
- Visual Studio Code is a general-purpose Code Editor for almost any kind of programming language or framework through so-called extensions that you can install on top of the IDE.

PhpStorm

IDE for PHP development by JetBrains



The screenshot shows the PhpStorm website landing page. At the top, the JetBrains logo is on the left, and navigation links for Developer Tools, Team Tools, Education, Solutions, Support, and Store are on the right. Below the navigation, there are links for 'Coming in 2025.1', 'What's New', 'Features', 'Resources', 'Pricing', and a 'Download' button. The main content area features the text 'PhpStorm Empowering PHP Developers' in large white font, with 'Code smarter with the PHP IDE' below it. A 'Download' button is present, with the text 'Free 30-day full trial No credit card required' next to it. At the bottom, it says 'Everything you need for Symfony'.

Free Educational Use:

- JetBrains offers free PhpStorm licenses and special deals for educational purposes.
- Students and teachers are eligible to use the JetBrains All Products Pack (which includes PhpStorm, as well as other JetBrains IDEs and tools) free of charge.

<https://www.halvorsen.blog>

Getting Started with PHP Programming



Hans-Petter Halvorsen

[Table of Contents](#)

Getting Started

- We need **web server with PHP installed**
 - You can setup your own server with PHP, but I will just use an existing webserver with PHP in this tutorial.
- We need a Code Editor like **Visual Studio Code**.
- We need an **FTP** program like **WinSCP** to transfer files from local PC to the webserver

My first PHP page

Visual Studio Code

```
<?php
$name = "Hans-Petter Halvorsen";
echo "Hello, my name is $name";
?>
```

index.php

PHP Files have the file extension **.php**

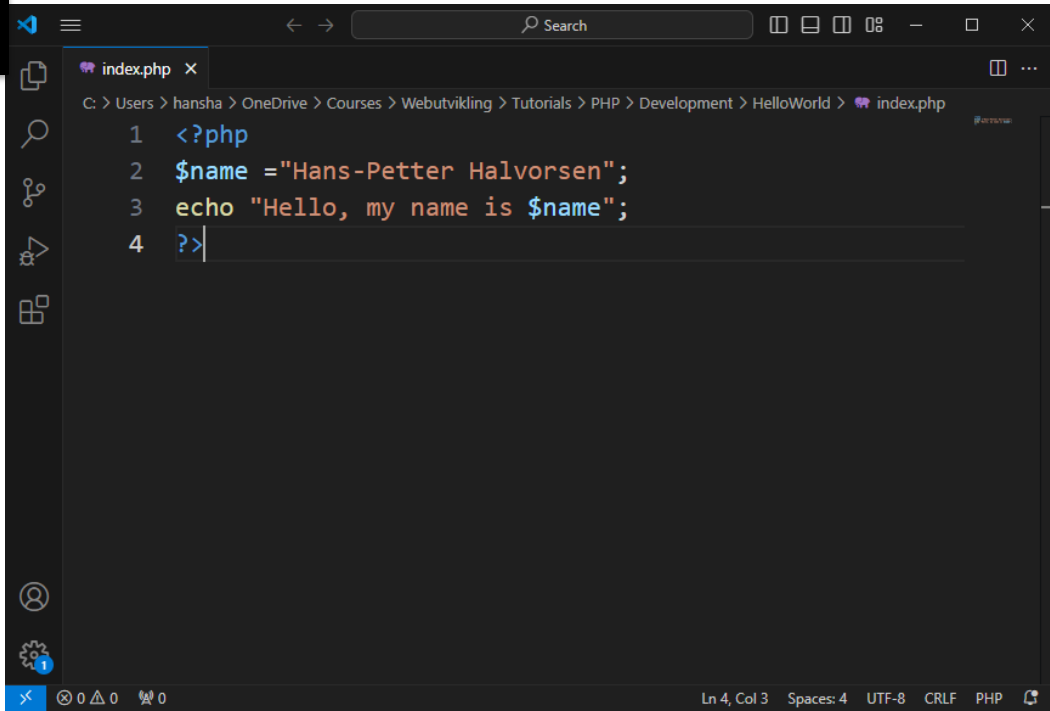
The startup file is typically names **index.php**

PHP code should be put inside

```
<?php
```

```
..
?>
```

Variables starts with **\$**



```
index.php x
C: > Users > hansha > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > HelloWorld > index.php
1 <?php
2 $name = "Hans-Petter Halvorsen";
3 echo "Hello, my name is $name";
4 ?>
```

Upload File using FTP - WinSCP

The screenshot displays the WinSCP interface with two panels. The left panel shows the local file system at `C:\Users\hansha\OneDrive\Courses\Webutvikling\Tutorials\PHP\Development\HelloWorld\`. The right panel shows the remote FTP directory at `/home/hansha/public_html/`. A text overlay is centered on the screen.

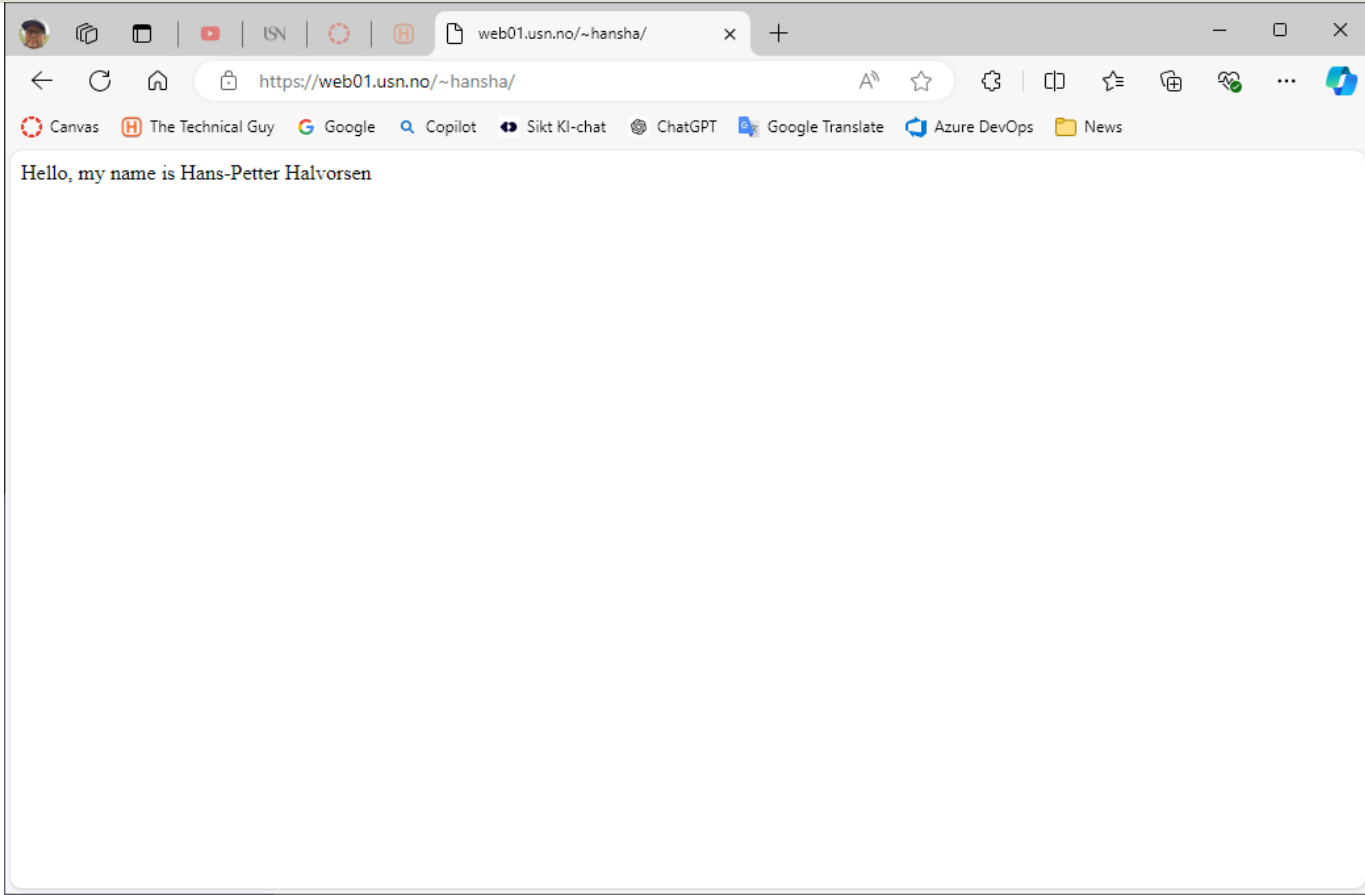
Here I have used the **WinSCP** FTP software

Name	Size	Type	Changed	Rights	Owner
..		Parent directory	03.09.2024 11:05:56		
index.php	1 KB	PHP Source File	03.09.2024 11:09:08		

Name	Size	Changed	Rights	Owner
		20.02.2024 15:01:39	rw-r-x--x	hansha

75 B of 75 B in 1 of 1 0 B of 0 B in 0 of 0 SFTP-3 0:05:37

Test in your Web Browser



Code Explained

```
<?php
$name = "Hans-Petter Halvorsen";
echo "Hello, my name is $name";
?>
```

PHP code should be put inside

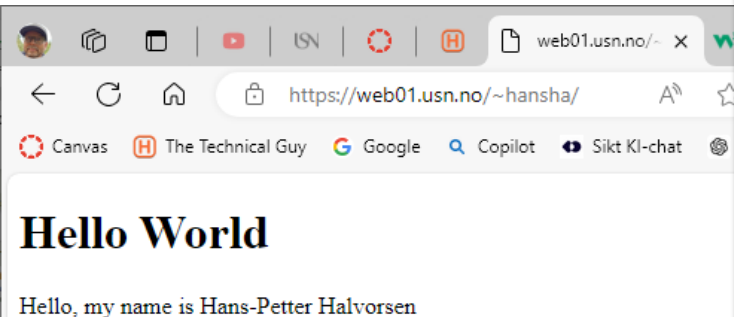
```
<?php
..
?>
```

Variables starts with **\$**

“**echo**” is a built-in function in PHP that is much used to output text or contents of a variable to the web browser.

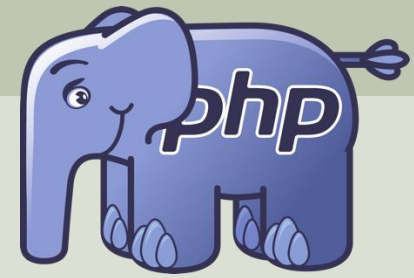
HTML + PHP

Typically, you include PHP code in between your HTML code. Here is a basic example:



```
<!DOCTYPE html>
<html>
  <body>
    <h1>Hello World</h1>
    <?php
      $name = "Hans-Petter Halvorsen";
      echo "Hello, my name is $name";
    ?>
  </body>
</html>
```

<https://www.halvorsen.blog>



PHP Programming



Hans-Petter Halvorsen

[Table of Contents](#)

PHP

PHP is a programming language with all the functionalities that a standard programming language has, like:

- Variables, Data Types, Arrays, etc.
- If.. Else.., While Loops, For Loops, etc.
- Functions, Classes and OOP.
- In addition, PHP has lots of web specific functionality.
- PHP has also built in support for Databases, i.e., inserting and retrieving data from different database systems.

Here we will focus on the specific web features and database features, and not plain programming, since this is like C++/C# and any other programming language.

Variables in PHP

- Variables in PHP starts with **\$**
- “**echo**” is a built-in function in PHP that is much used to output contents of a variable to the web browser. You can also use the **print()** function.
- Note! Variables in PHP is case-sensitive!
- You don't need to define the datatype of the variable in PHP (PHP is a so-called loosely typed language)

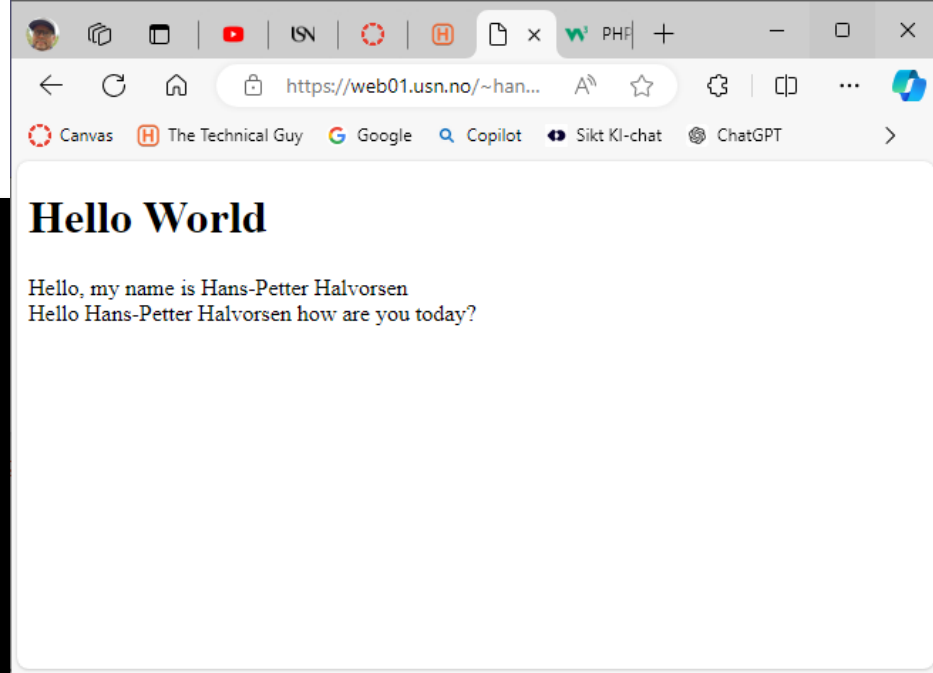
Combining Text and Variables

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Hello World</h1>
    <?php
      $name = "Hans-Petter Halvorsen";

      echo "Hello, my name is $name";

      echo "<br>";

      echo 'Hello ' . $name . ' how are you today?';
    ?>
  </body>
</html>
```



Combining Text and Variables

There is a huge difference between **double quotes** (") and **single quotes** (') in PHP:

Here you see some examples:

```
$name ="Hans-Petter Halvorsen";  
echo "Hello, my name is $name";
```

Double quotes ("):

Here will \$name be treated as a variable

```
$name ="Hans-Petter Halvorsen";  
echo 'Hello, my name is $name';
```

Single quotes ('):

Here will \$name just be part of the string

```
$name ="Hans-Petter Halvorsen";  
  
echo 'Hello ' . $name . ' how are you today?';
```

If using single quotes ('), you can use **.\$name.** to treat \$name as a variable

```
File Edit Selection View Go Run Terminal Help
index.php x
C: > Users > hansha > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > HelloWorld > index.php
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Hello World</h1>
5     <?php
6       $name = "Hans-Petter Halvorsen";
7
8       echo "Hello, my name is $name";
9
10      echo "<br>";
11
12      echo 'Hello ' . $name . ' how are you today?';
13    ?>
14  </body>
15 </html>
```

1

PHP code server-side

Final Result shown in the Web Browser

Hello World

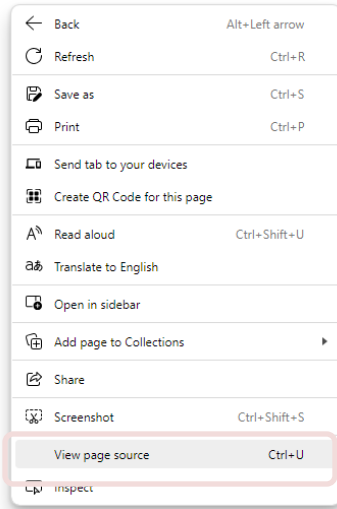
3

Hello, my name is Hans-Petter Halvorsen
Hello Hans-Petter Halvorsen how are you today?

```
Line wrap
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Hello World</h1>
5     Hello, my name is Hans-Petter Halvorsen<br>Hello Hans-Petter Halvorsen how are you today?
6   </body>
7 </html>
```

2

Generated HTML File that is sent to the Client (Web Browser)



String Function in PHP

PHP has many useful built-in functions for string manipulation, e.g.:

- `strlen()`
- `str_word_count()`
- `str_replace()`
- `strpos()`
- `substr()`
- `strrev()`
- `trim()`
- `strtoupper()`
- `strtolower()`
- `+++`

<https://www.php.net/manual/en/ref.strings.php>

Comments in PHP

```
..  
  
// Single line comment  
  
/* Multiline comment  
..  
..  
..  
*/
```

It is good practice to add and use comments inside your code

Using comments are also a good “Debugging technique” by commenting out one or more code lines and make those are not executed.

If .. Else, Loops, etc.

PHP has built-in functionality for Conditions and Loops as other programming languages. PHP supports many different types, here you see some basic example. The syntax is very similar to C/C#.

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Getting Started with PHP</h1>
    <?php
      $number = 18;

      if ($number > 10)
      {
        echo "The number is larger than 10";
      }
      else
      {
        echo "The number is smaller than 10";
      }
    ?>
  </body>
</html>
```

Functions

PHP has many useful built-in function, but you can of course also make your own functions

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Getting Started with PHP</h1>
    <?php
      function FindAverage($number1, $number2)
      {
          $average = ($number1 + $number2)/2;
          return $average;
      }

      $x = 6;
      $y = 12;
      $mean = FindAverage($x, $y);
      echo $mean
    ?>
  </body>
</html>
```


<https://www.halvorsen.blog>

HTML Forms in PHP



Hans-Petter Halvorsen

[Table of Contents](#)

HTML Forms

- An HTML form is typically used to collect data from the user.
- Then typically the data will then be sent to the server for processing and storage.

```
<!DOCTYPE html>
<html>
<body>

<h1>Please enter your User Data</h1>

<form action="showdata.php" method="POST">
  <label for="firstName">First Name:</label><br>
  <input type="text" id="firstName" name="firstName">
  <br>

  <label for="lastName">Last Name:</label><br>
  <input type="text" id="lastName" name="lastName">
  <br><br>

  <input type="submit" value="Save">
</form>

</body>
</html>
```

Forms and POST Example

```
<!DOCTYPE html>
<html>
<body>

<h1>Please enter your User Data</h1>

<form action="showdata.php" method="POST">
  <label for="firstName">First Name:</label><br>
  <input type="text" id="firstName" name="firstName">
  <br>

  <label for="lastName">Last Name:</label><br>
  <input type="text" id="lastName" name="lastName">
  <br><br>

  <input type="submit" value="Save">
</form>

</body>
</html>
```

Please enter your User Data

First Name:

Last Name:

Angular Snip

```
<!DOCTYPE html>
<html>
<body>

<h1>Information about your User Data</h1>

Your First Name is <?php echo $_POST["firstName"]; ?>
<br>

Your Last Name is: <?php echo $_POST["lastName"]; ?>

</body>
</html>
```

showdata.php

Information about your User Data

Your First Name is Hans Petter
Your Last Name is: Halvorsen

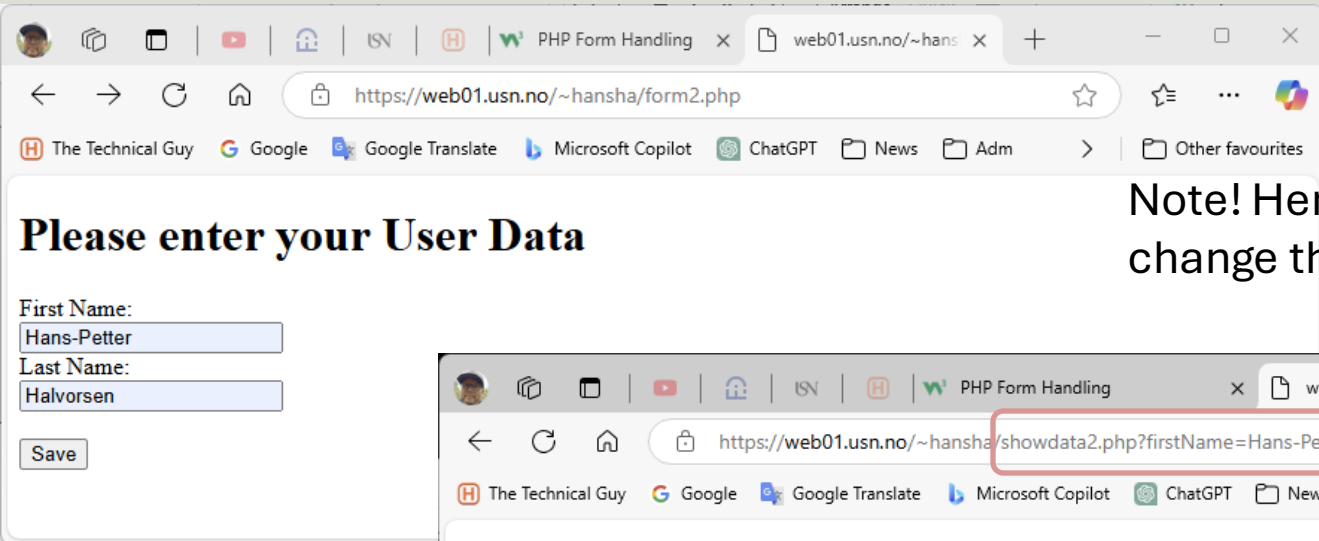
We use **\$_POST** to get the Form Data

GET Example

```
form2.php x showdata2.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > HelloWorld > form2.php
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Please enter your User Data</h1>
6
7 <form action="showdata2.php" method="GET">
8   <label for="firstName">First Name:</label><br>
9   <input type="text" id="firstName" name="firstName">
10  <br>
11
12   <label for="lastName">Last Name:</label><br>
13   <input type="text" id="lastName" name="lastName">
14   <br><br>
15
16   <input type="submit" value="Save">
17 </form>
18
19
20 </body>
21 </html>
```

```
form2.php x showdata2.php x
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > HelloWorld > showdata2.php
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Information about your User Data</h1>
6
7 Your First Name is <?php echo $_GET["firstName"]; ?>
8 <br>
9
10 Your Last Name is: <?php echo $_GET["lastName"]; ?>
11
12 </body>
13 </html>
```

GET and Query String



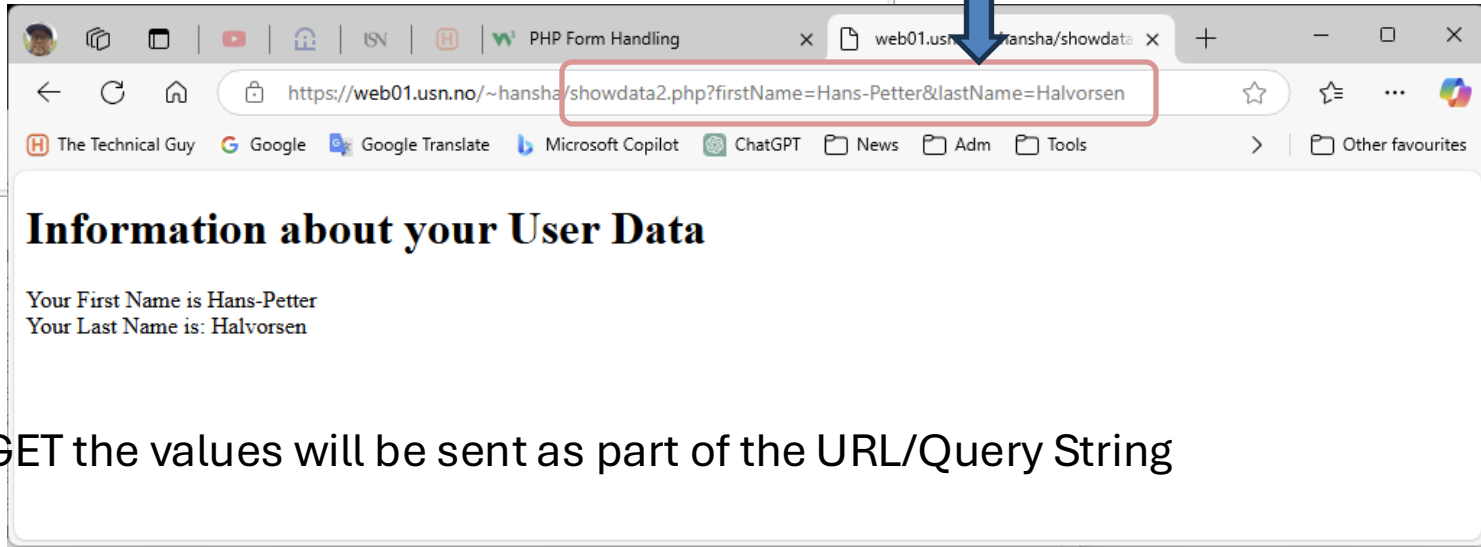
Please enter your User Data

First Name:
Hans-Petter

Last Name:
Halvorsen

Save

Note! Here we can also manually change the Query String Data.



Information about your User Data

Your First Name is Hans-Petter
Your Last Name is: Halvorsen

When using GET the values will be sent as part of the URL/Query String

<https://www.halvorsen.blog>

Session Variables



Hans-Petter Halvorsen

[Table of Contents](#)

Session Variables

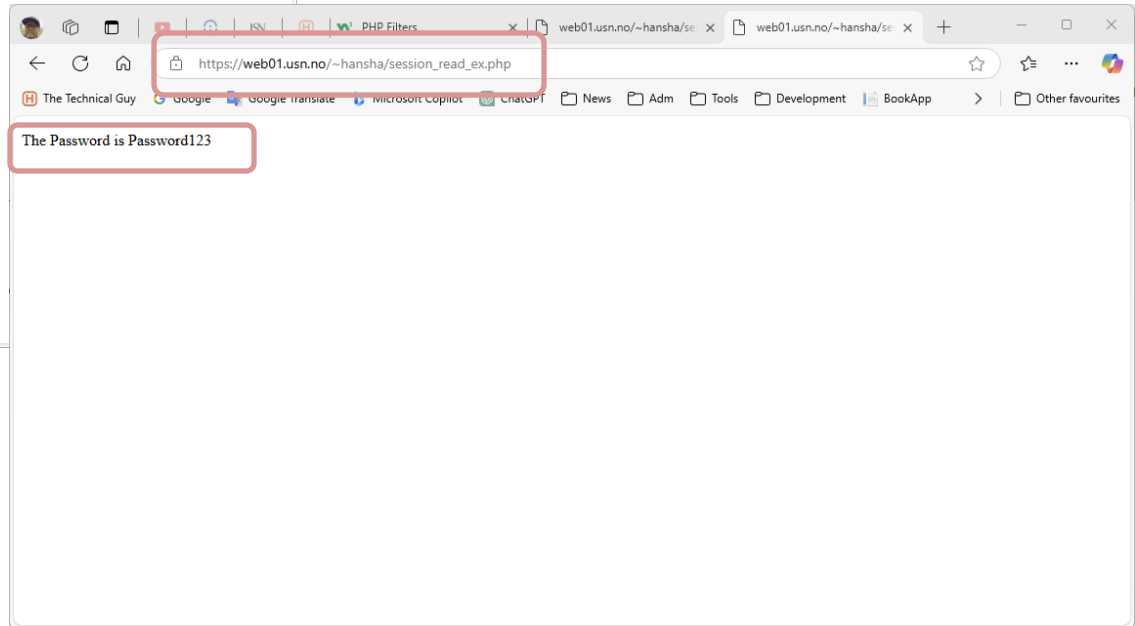
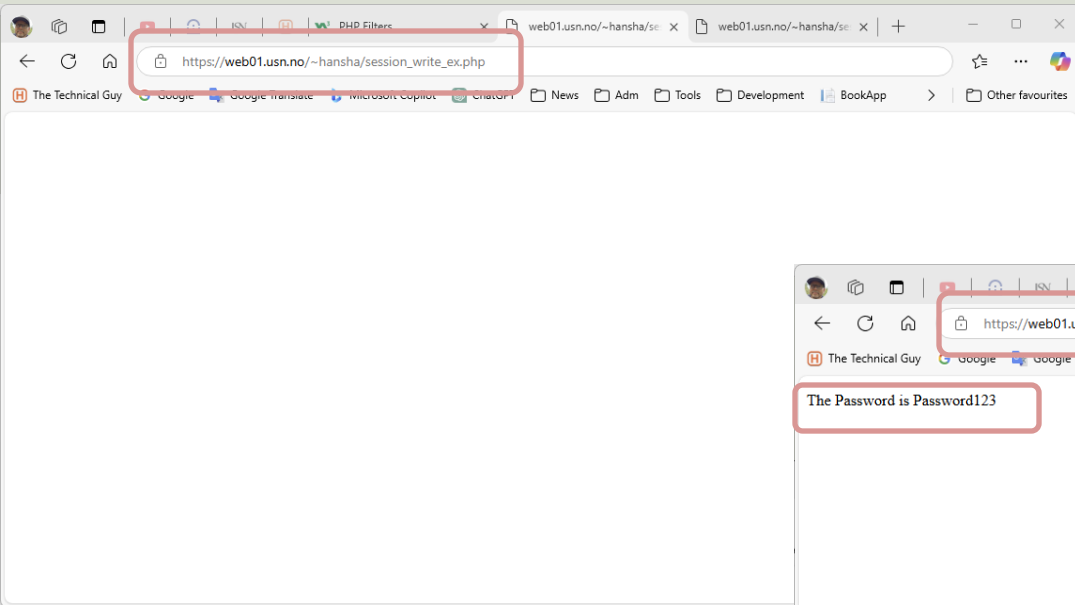
- We can use something called Session variables in order to send data between 2 web pages.
- Unlike a cookie, the information is not stored on the users PC.
- Session variables are very handy in web development.
- Session variables hold information only for one single user (you), so this means the information is only available for you and not for other users of the web page or web application.
- All web frameworks has these Sessions variables.

Session Example

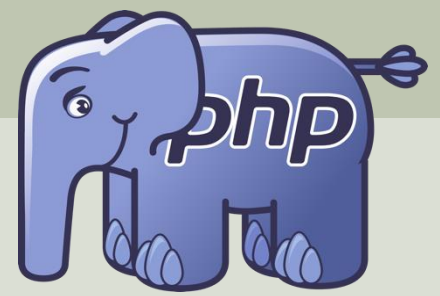
```
File Edit Selection View ... Search
session_write_ex.php session_read_ex.php
C:\Users> hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Hello
1 <?php
2 session_start();
3 ?>
4
5 <!DOCTYPE html>
6 <html>
7 <body>
8
9 <?php
10 $_SESSION["password"] = "Password123";
11 ?>
12
13 </body>
14 </html>
```

```
File Edit Selection View ... Search
session_write_ex.php session_read_ex.php
C:\Users> hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > HelloWorld > session_read_ex.php
1 <?php
2 session_start();
3 $password = $_SESSION["password"];
4 ?>
5
6 <!DOCTYPE html>
7 <html>
8 <body>
9
10 <p>
11 <?php
12 echo "The Password is $password";
13 ?>
14 </p>
15
16 </body>
17 </html>
```


Session Example



<https://www.halvorsen.blog>



PHP and MySQL Database



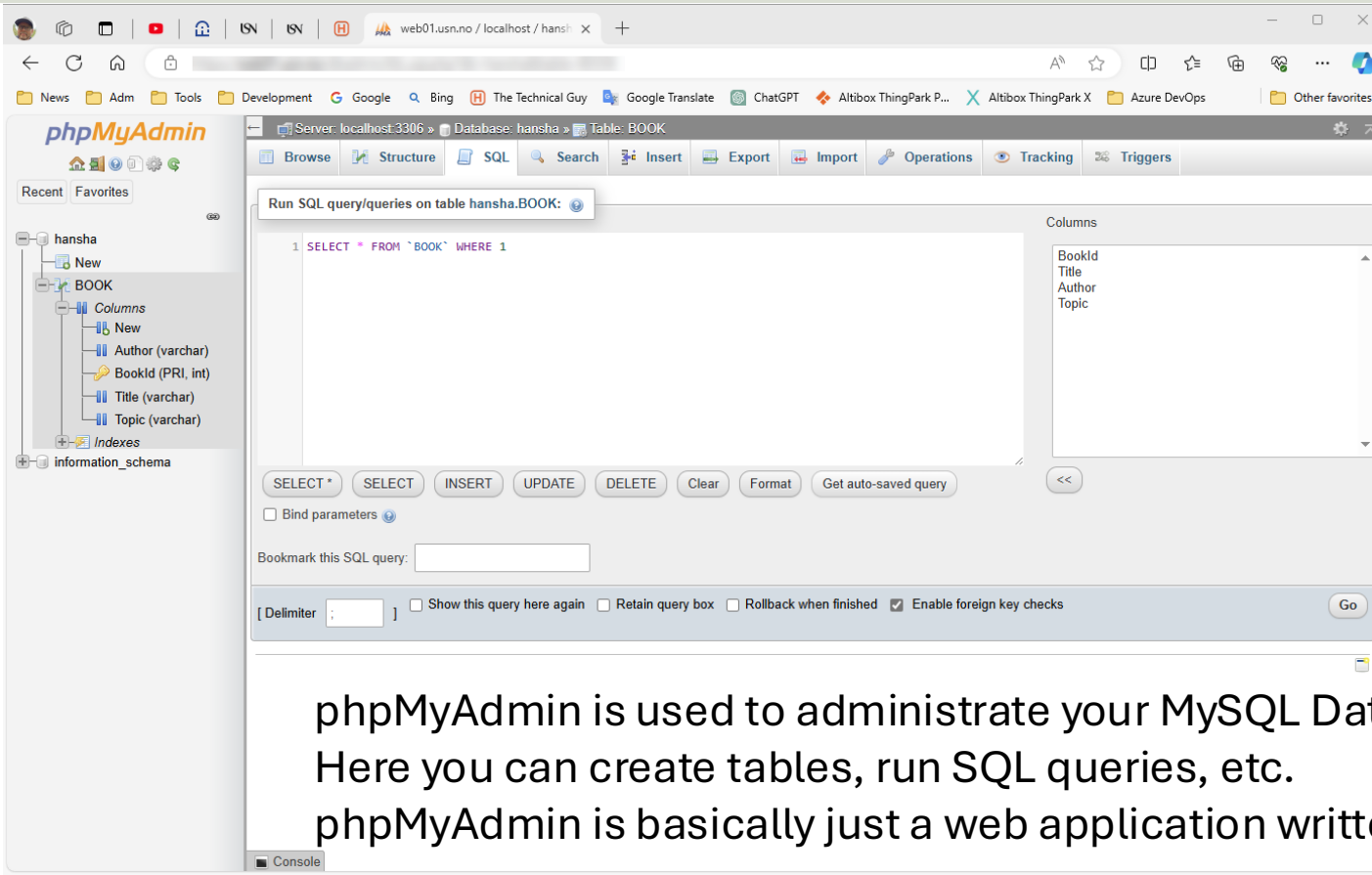
Hans-Petter Halvorsen

[Table of Contents](#)

MySQL and PHP

- MySQL is a popular relational database system.
- It is free and open source.
- MySQL uses SQL (Structured Query Language)
- The combination of PHP and MySQL is very popular.
- You can create powerful web applications where you can show, save, update and delete data in a MySQL database from the PHP code.

phpMyAdmin



The screenshot displays the phpMyAdmin interface in a web browser. The browser's address bar shows the URL `web01.usn.no / localhost / hansha`. The phpMyAdmin interface includes a navigation sidebar on the left with a tree view showing the database structure: `hansha` (database), `New` (table), `BOOK` (table), `Columns` (sub-section), `New` (button), `Author (varchar)`, `BookId (PRI, int)`, `Title (varchar)`, `Topic (varchar)`, `Indexes`, and `information_schema`. The main content area is titled "Run SQL query/queries on table hansha.BOOK:" and features a toolbar with options: `Browse`, `Structure`, `SQL`, `Search`, `Insert`, `Export`, `Import`, `Operations`, `Tracking`, and `Triggers`. The SQL query editor contains the text: `1 SELECT * FROM `BOOK` WHERE 1`. To the right of the editor is a "Columns" list containing: `BookId`, `Title`, `Author`, and `Topic`. Below the editor are buttons for `SELECT *`, `SELECT`, `INSERT`, `UPDATE`, `DELETE`, `Clear`, `Format`, and `Get auto-saved query`. There is also a checkbox for `Bind parameters` and a "Bookmark this SQL query:" input field. At the bottom, there are options for `[Delimiter :]`, `Show this query here again`, `Retain query box`, `Rollback when finished`, `Enable foreign key checks`, and a `Go` button.

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.

Connect to the Database

There are 2 different methods that you can use to connect to your MySQL Database from PHP:

- MySQLi – Only works together with MySQL
- PDO – This option will also work for many other types of database systems.

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);
```

CRUD

Typically, we want do the following operations:

- **C**reate (Insert) Data
- **R**ead (Select) Data
- **U**ppdate Data
- **D**eleete Data

=> This is referred to as CRUD

Typically, all Applications today need to communicate with a Database and has CRUD functionality.

When you have learned to create a basic CRUD Application, you have all the necessary tools you need to create any kind of Application.

SQL

- Structured Query Language (SQL) is used to write, read and update data from Database Systems.
- SQL is a standardized language used by most database systems.
- You can use SQL inside the “SQL Server Management Studio” or inside your C# App.
- SQL Example: `select * from SCHOOL`

SQL Query Examples

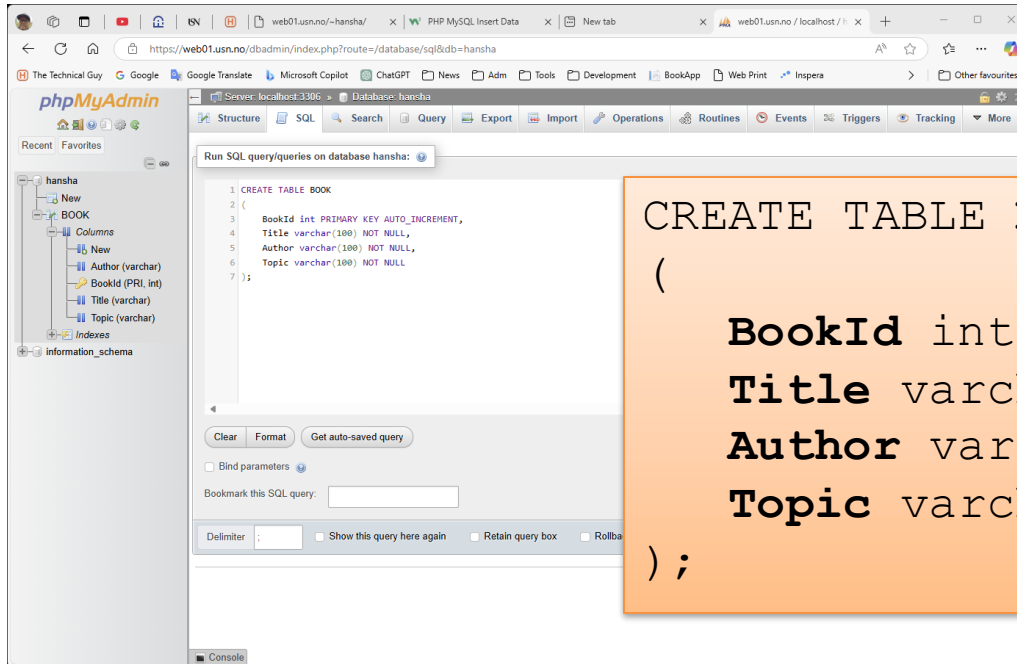
- **insert** into STUDENT (Name , Number, SchoolId)
values ('John Smith', '100005', 1)
- **select** SchoolId, Name from SCHOOL
- **select** * from SCHOOL where SchoolId > 100
- **update** STUDENT set Name='John Wayne' **where** StudentId=2
- **delete** from STUDENT **where** SchoolId=3

We have 4 different Query Types: **INSERT**, **SELECT**, **UPDATE** and **DELETE**

CRUD: **C** – Create or Insert Data, **R** – Retrieve (Select) Data, **U** – Update Data, **D** – Delete Data

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.



```
CREATE TABLE BOOK
(
  BookId int PRIMARY KEY AUTO_INCREMENT,
  Title varchar(100) NOT NULL,
  Author varchar(100) NOT NULL,
  Topic varchar(100) NOT NULL
);
```

Database

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

```
insert into BOOK (Title, Author, Topic) values  
( 'Web Apps', 'Elvis Presly', 'Programming');
```

```
insert into BOOK (Title, Author, Topic) values  
( 'IoT and Cloud', 'John Wayne', 'IoT');
```

```
insert into BOOK (Title, Author, Topic) values  
( 'C#', 'Rune Hansen', 'Programming');
```

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.

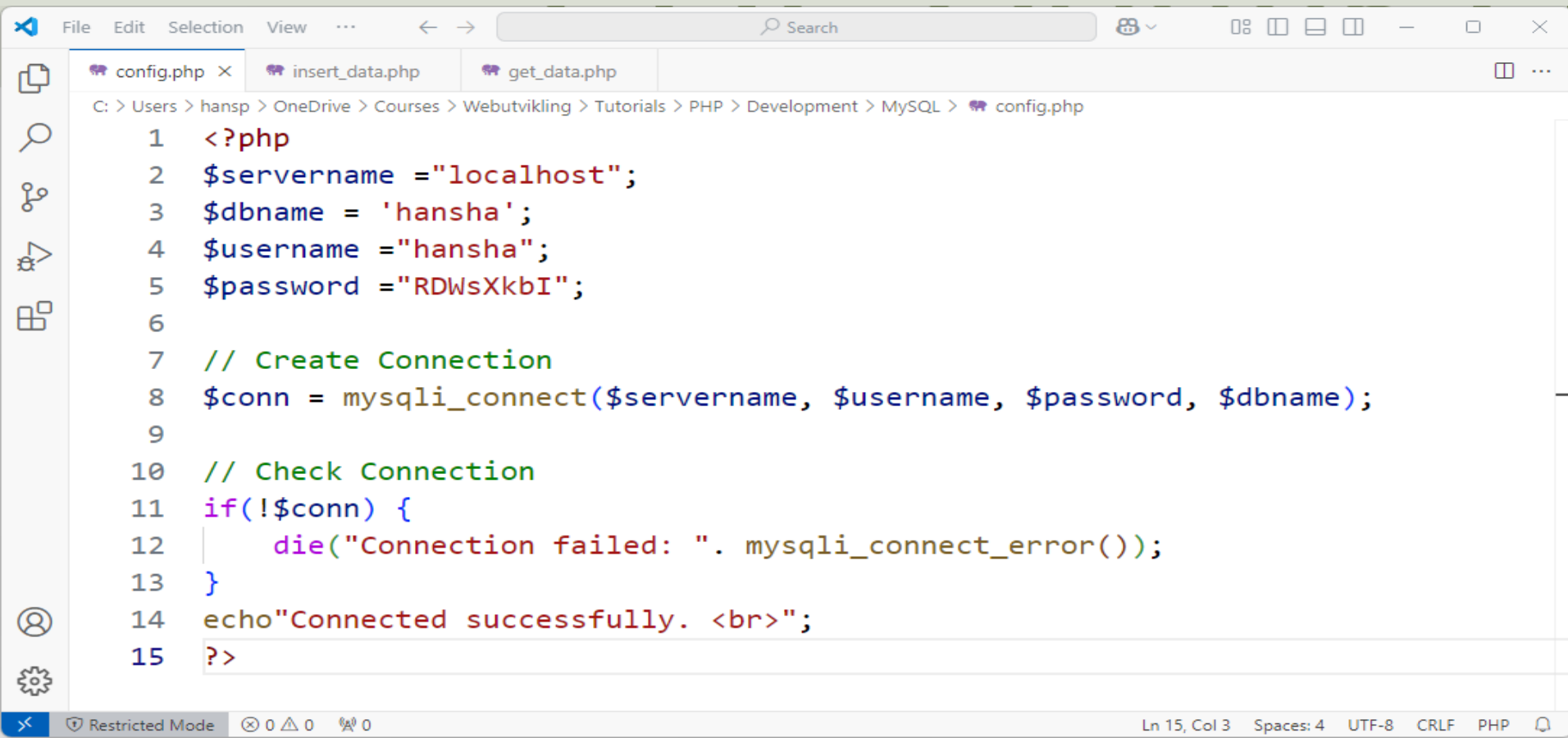
```
<?php
$servername = "localhost";
$dbname = 'hansha';
$username = "hansha";
$password = "RDWsXkbI";

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

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

config.php

PHP Config File



The image shows a code editor window with a light green background. The title bar at the top contains the text "File Edit Selection View" followed by navigation arrows and a search box. Below the title bar, there are three tabs: "config.php", "insert_data.php", and "get_data.php". The main editor area displays the following PHP code:

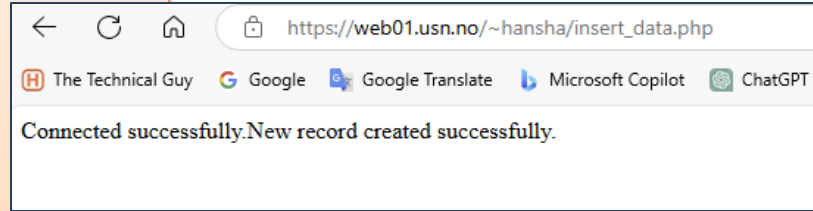
```
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > MySQL > config.php
1  <?php
2  $servername = "localhost";
3  $dbname = 'hansha';
4  $username = "hansha";
5  $password = "RDWsXkbI";
6
7  // Create Connection
8  $conn = mysqli_connect($servername, $username, $password, $dbname);
9
10 // Check Connection
11 if(!$conn) {
12     die("Connection failed: ". mysqli_connect_error());
13 }
14 echo"Connected successfully. <br>";
15 ?>
```

At the bottom of the editor, there is a status bar with the text "Restricted Mode" and several icons. On the right side of the status bar, it shows "Ln 15, Col 3", "Spaces: 4", "UTF-8", "CRLF", "PHP", and a refresh icon.

Save Data to the Database

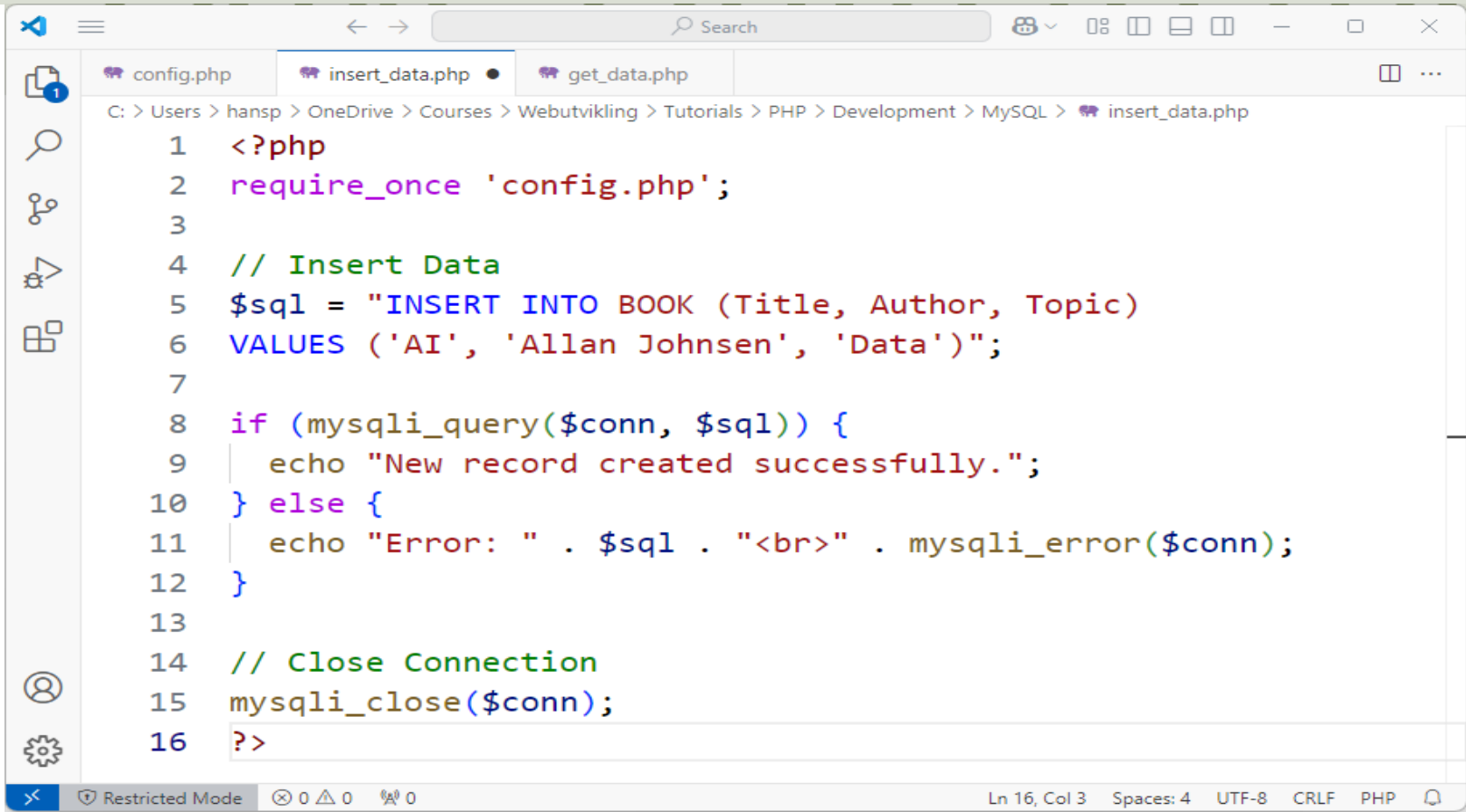
```
<?php
require_once 'config.php';
// Insert Data
$sql = "INSERT INTO BOOK (Title, Author, Topic)
VALUES ('AI', 'Allan Johnsen', 'Data')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully.";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
// Close Connection
mysqli_close($conn);
?>
```



Then go to **phpMyAdmin** and check if the data has been stored in the database.

Save Data to the Database



The image shows a code editor window with three tabs: config.php, insert_data.php (selected), and get_data.php. The file path is C:\Users> hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > MySQL > insert_data.php. The code is as follows:

```
1 <?php
2 require_once 'config.php';
3
4 // Insert Data
5 $sql = "INSERT INTO BOOK (Title, Author, Topic)
6 VALUES ('AI', 'Allan Johnsen', 'Data')";
7
8 if (mysqli_query($conn, $sql)) {
9     echo "New record created successfully.";
10 } else {
11     echo "Error: " . $sql . "<br>" . mysqli_error($conn);
12 }
13
14 // Close Connection
15 mysqli_close($conn);
16 ?>
```

The status bar at the bottom shows "Restricted Mode", "0 0 0", "Ln 16, Col 3", "Spaces: 4", "UTF-8", "CRLF", "PHP", and a search icon.

Show Data from the Database

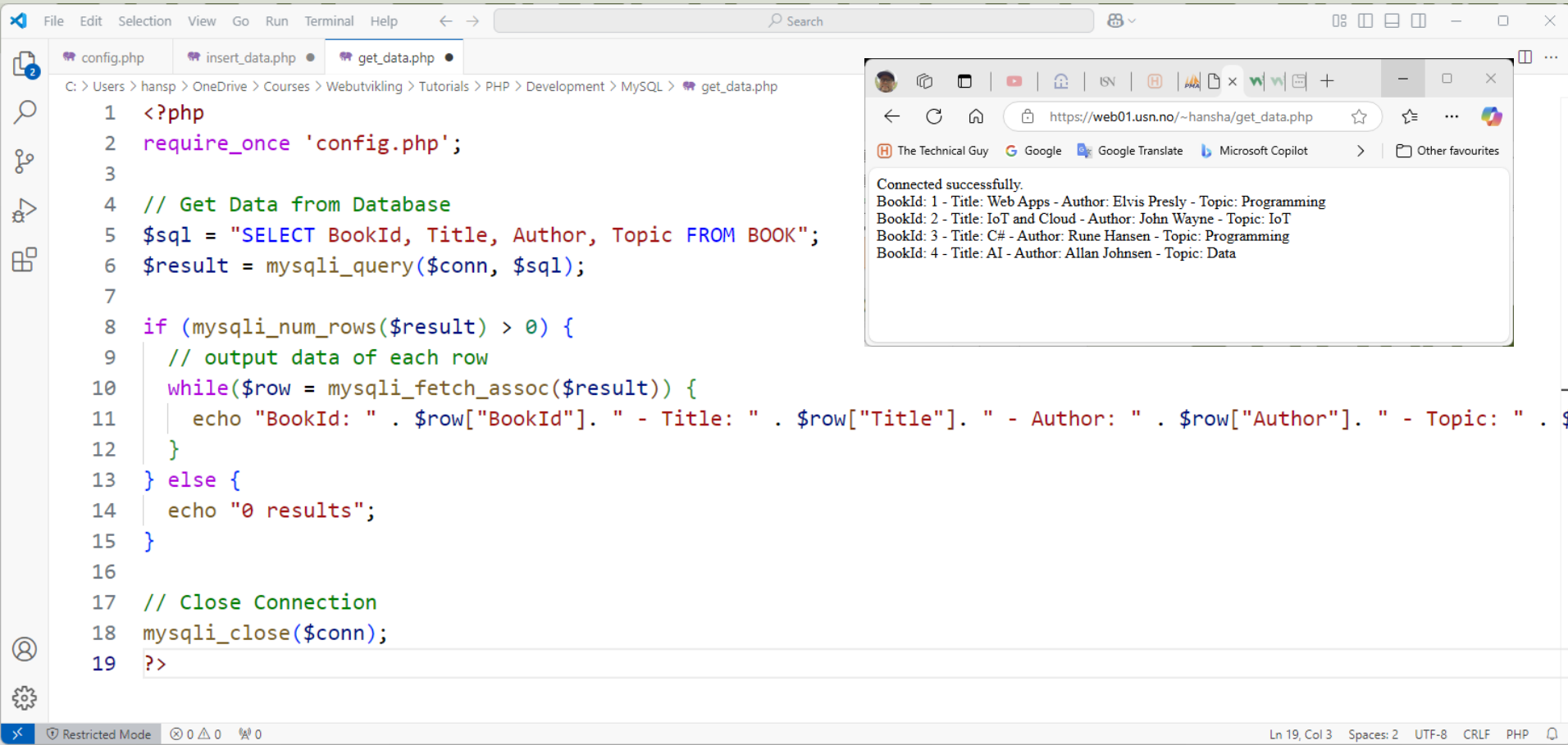
```
<?php
require_once 'config.php';

// Get Data from Database
$sql = "SELECT BookId, Title, Author, Topic FROM BOOK";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo "BookId: " . $row["BookId"]. " - Title: " . $row["Title"]. " - Author: " .
$row["Author"]. " - Topic: " . $row["Topic"]. "<br>";
    }
} else {
    echo "0 results";
}

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


Show Data from the Database



The image shows a code editor window with a PHP script named `get_data.php`. The script connects to a MySQL database, queries a table named `BOOK`, and displays the results. The browser window shows the output of the script, which is a list of four books with their IDs, titles, authors, and topics.

```
1 <?php
2 require_once 'config.php';
3
4 // Get Data from Database
5 $sql = "SELECT BookId, Title, Author, Topic FROM BOOK";
6 $result = mysqli_query($conn, $sql);
7
8 if (mysqli_num_rows($result) > 0) {
9     // output data of each row
10    while($row = mysqli_fetch_assoc($result)) {
11        echo "BookId: " . $row["BookId"]. " - Title: " . $row["Title"]. " - Author: " . $row["Author"]. " - Topic: " . $row["Topic"];
12    }
13 } else {
14     echo "0 results";
15 }
16
17 // Close Connection
18 mysqli_close($conn);
19 ?>
```

Browser Output:

```
Connected successfully:
BookId: 1 - Title: Web Apps - Author: Elvis Presly - Topic: Programming
BookId: 2 - Title: IoT and Cloud - Author: John Wayne - Topic: IoT
BookId: 3 - Title: C# - Author: Rune Hansen - Topic: Programming
BookId: 4 - Title: AI - Author: Allan Johnsen - Topic: Data
```

Resources and References

- PHP Tutorial w3school:
<https://www.w3schools.com/php/>
- PHP Tutorial TutorialsPoint:
<https://www.tutorialspoint.com/php/>
- PHP Documentation:
<https://www.php.net/manual/en/>
- MySQL Tutorial:
<https://www.w3schools.com/mysql>

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: <https://www.halvorsen.blog>

