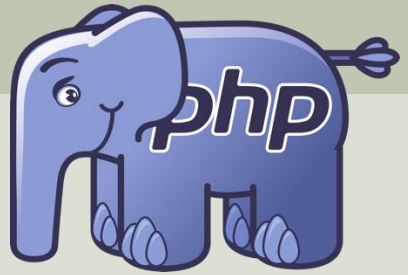
https://www.halvorsen.blog



PHP Tutorial

Hans-Petter Halvorsen

Contents

https://www.halvorsen.blog

PHP Tutorial

Introduction

Table of Contents

Hans-Petter Halvorsen

Web Development Frameworks

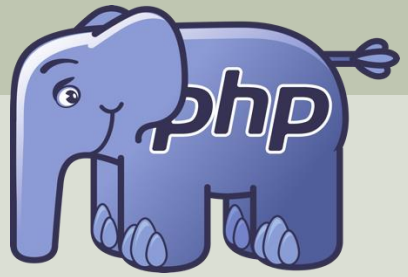
Some of the most used server-side (backend) Web Development Frameworks:

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

Tools

- **PHP** - a server scripting language for making dynamic web pages/applications, 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 system. MySQL is free and open-source.
- **phpMyAdmin** - a free and open-source administration tool for MySQL (and MariaDB).

https://www.halvorsen.blog



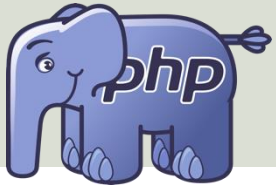
PHP Tutorial

Introduction to 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 + MySQL

- You need to have a PHP + MySQL Environment on your local computer on get access to it from a server/Internet.
- For local installation you need to download and install Apache (or another web server that supports PHP), PHP and MySQL.
- You can get server access from many providers (free or paid). Just search for, e.g., “PHP web hosting”.
- (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.

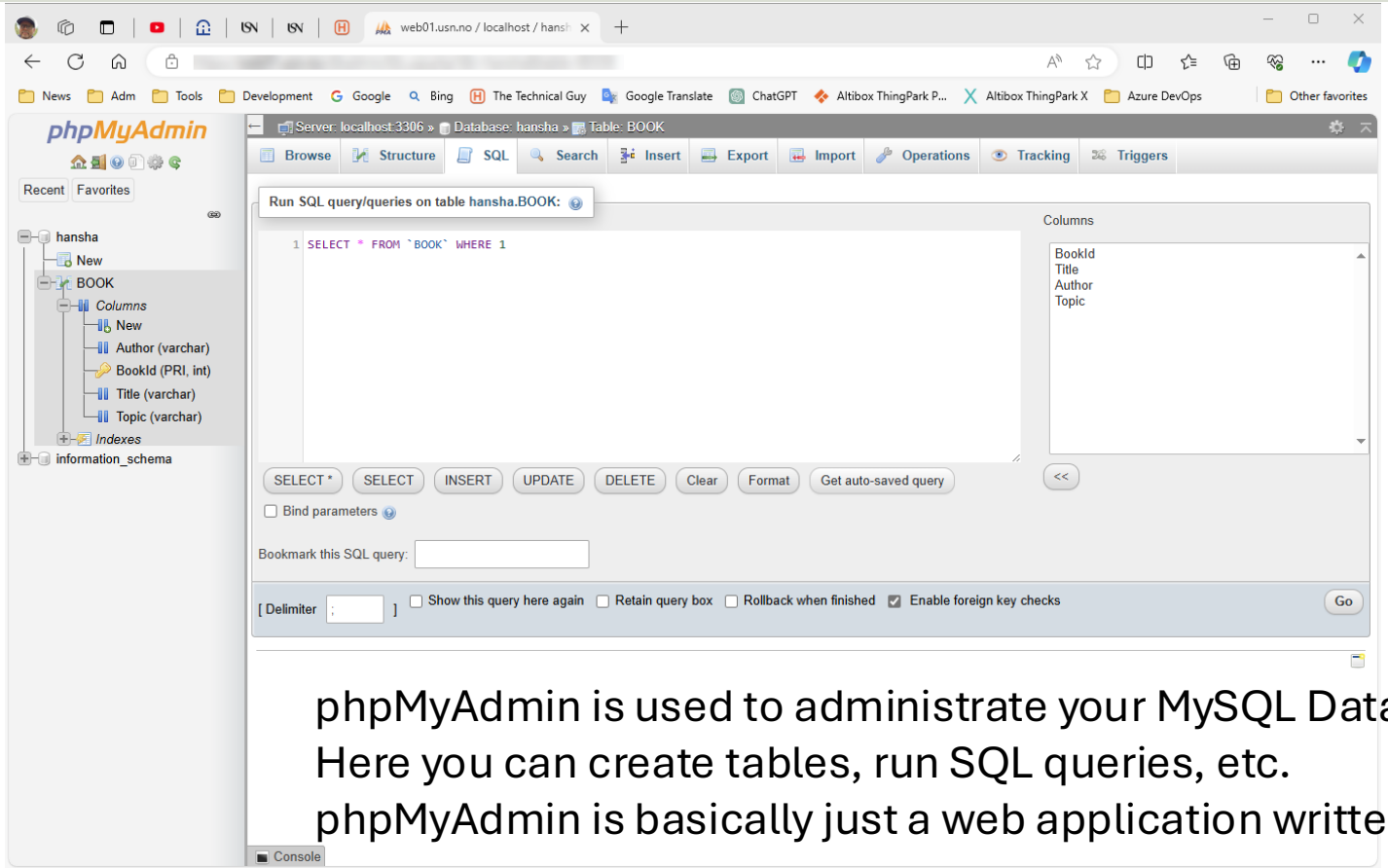
LAMP/PHP Web Hosting

- 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
- XAMPP is an acronym that stands for Cross-Platform, Apache, MySQL, PHP, and Perl.
- It works on Windows, macOS and Linux
- It installs Apache, MariaDB, PHP and Perl.
  - Apache is a Web Server.
  - MariaDB is almost identical to MySQL.
  - PHP and Perl are Web Programming languages.
- https://www.apachefriends.org

phpMyAdmin



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:

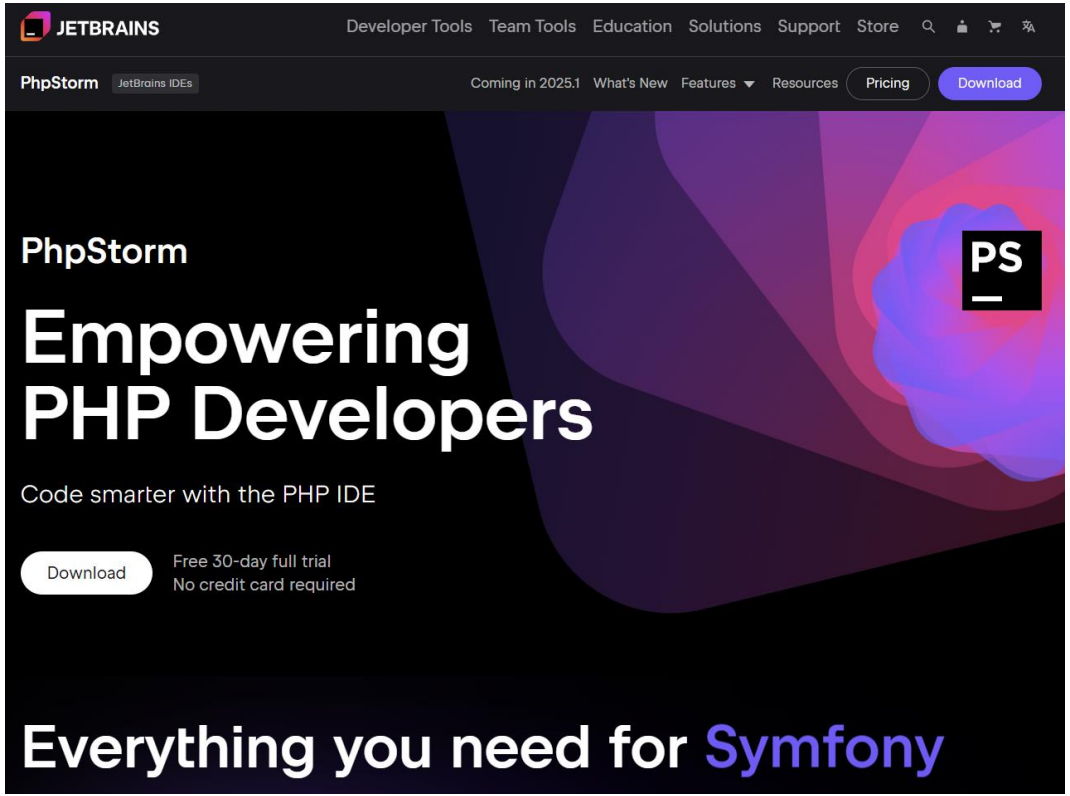
- Visual Studio Code.
  - Multiplatform and Free.
  - Homepage: https://code.visualstudio.com
- PhpStorm.
  - Free 30-day full trial.
  - Free access to all JetBrains IDEs for students and educators.
  - Homepage: https://www.jetbrains.com
- +++

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



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

PHP Tutorial

Getting Started with PHP

Hans-Petter Halvorsen

Table of Contents

Getting Started

- We need a 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.
- You will also need to have a MySQL Database for some parts in this tutorial.

My first PHP page

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

index.php

PHP Files have the file extension .php

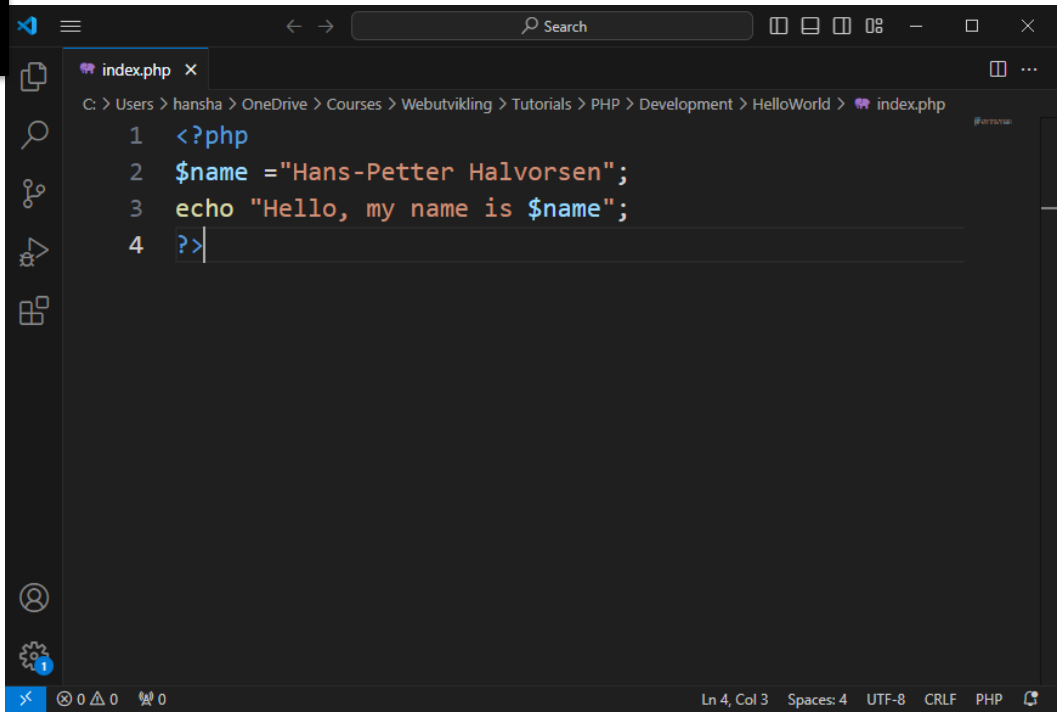
The startup file is typically names index.php

PHP code should be put inside

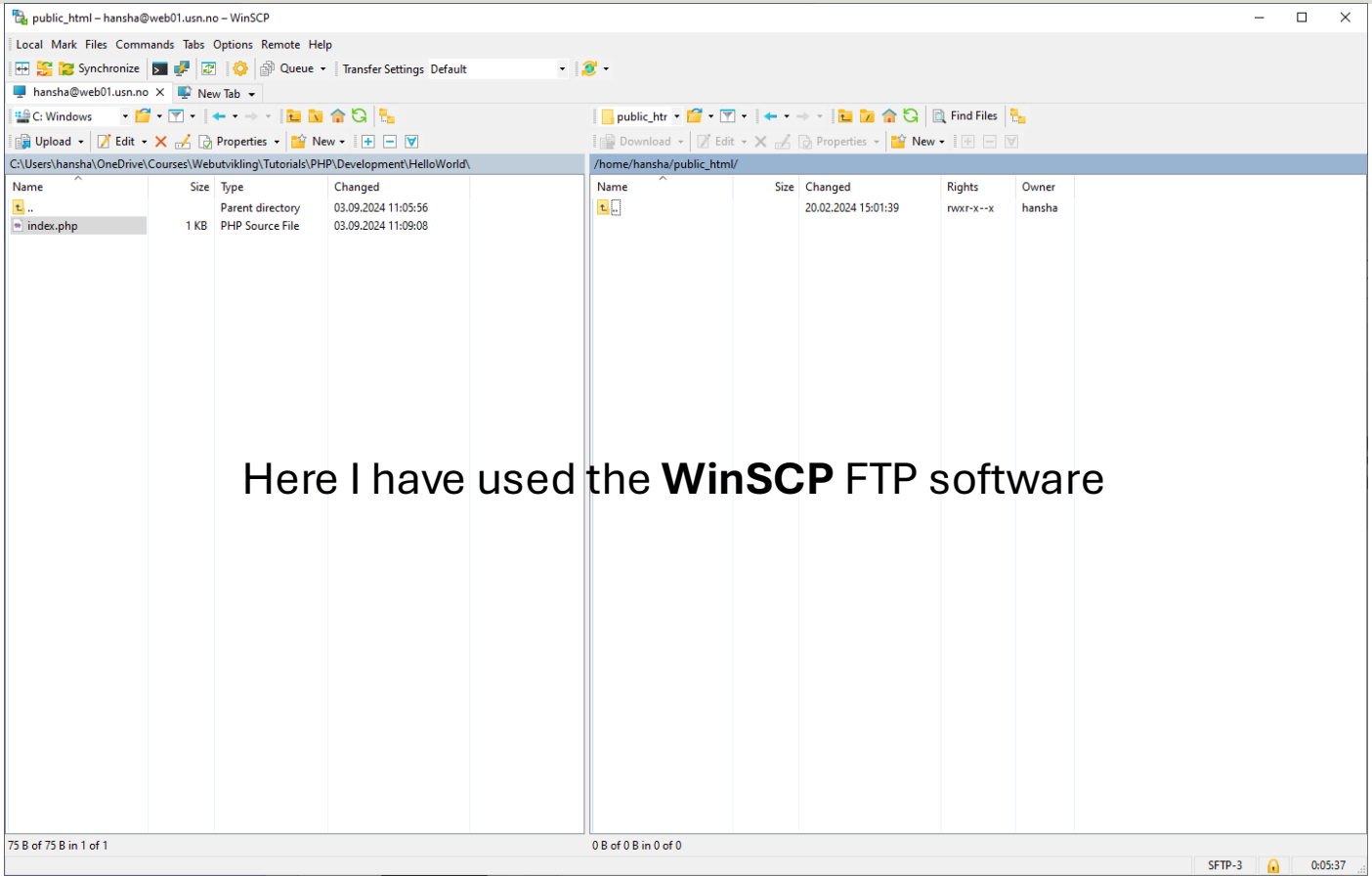
```
<?php
..
?>
```

Variables starts with $

Visual Studio Code

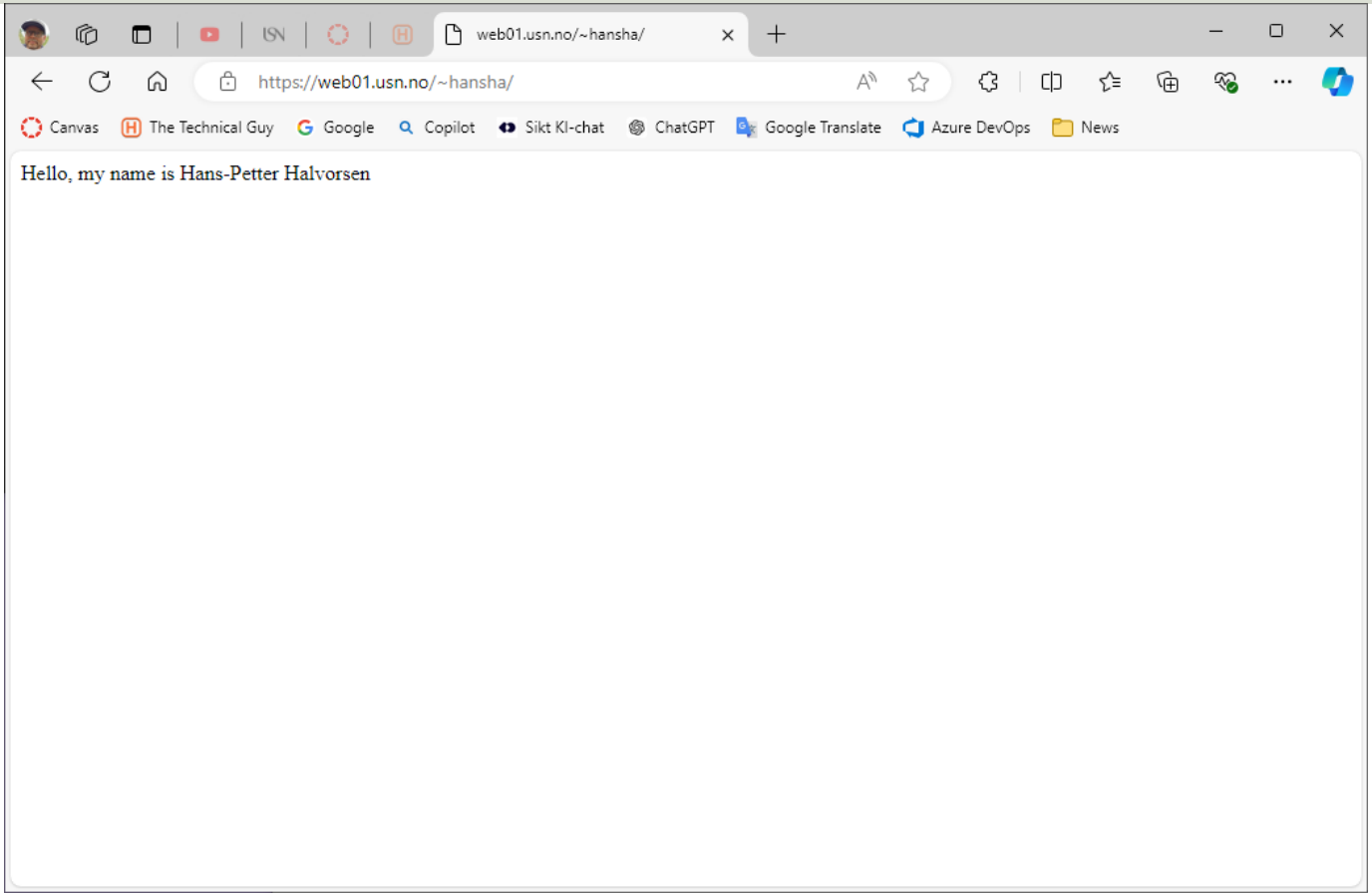


Upload File using FTP - **WinSCP**



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

Test in your Web Browser



Code Explained

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

PHP code should be put inside

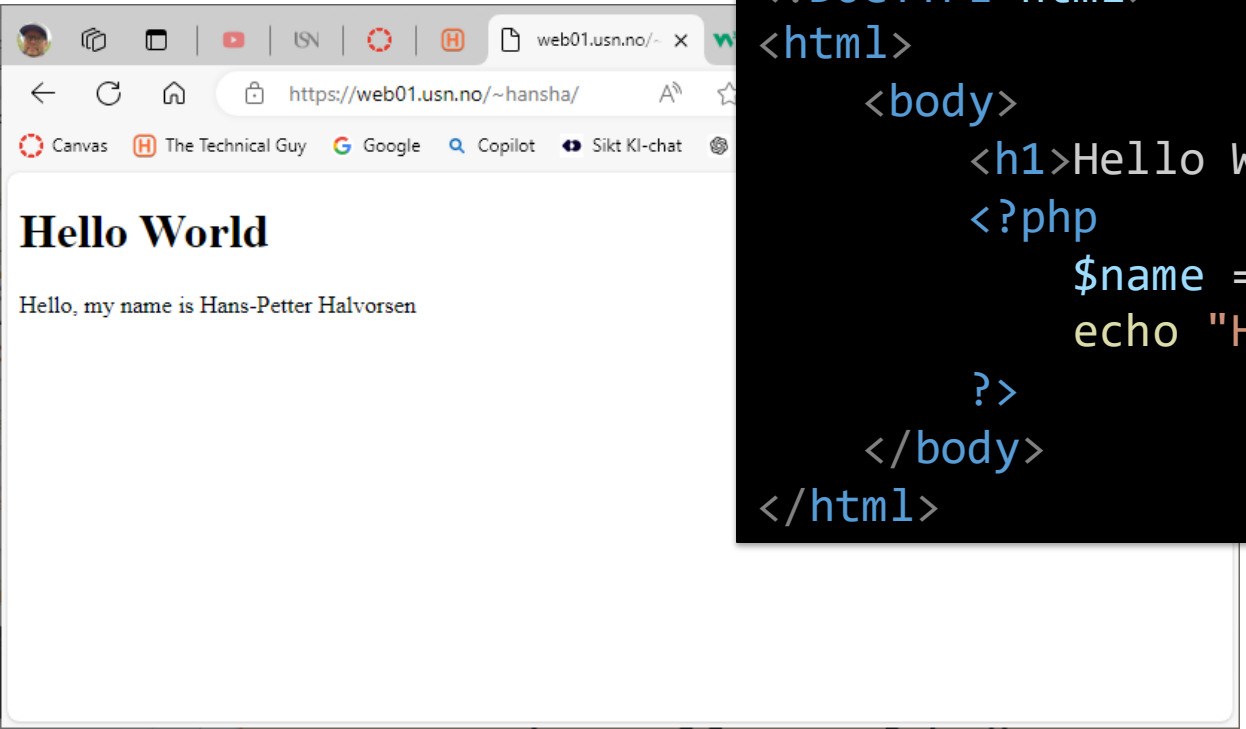
```
<?php
..
 ?>
```

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:




```php
<!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

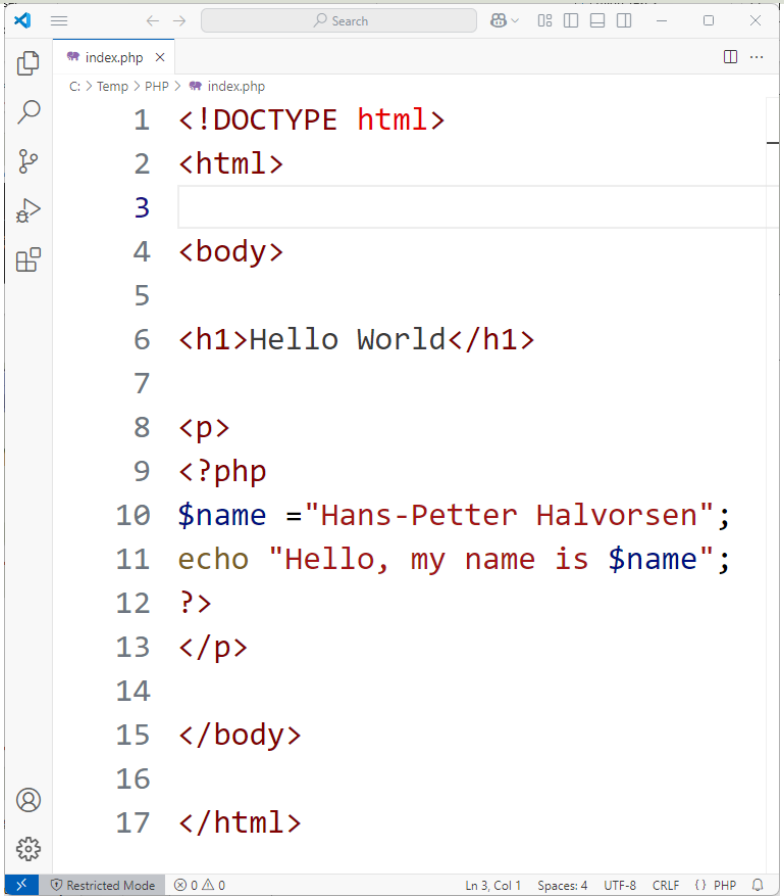
PHP Tutorial

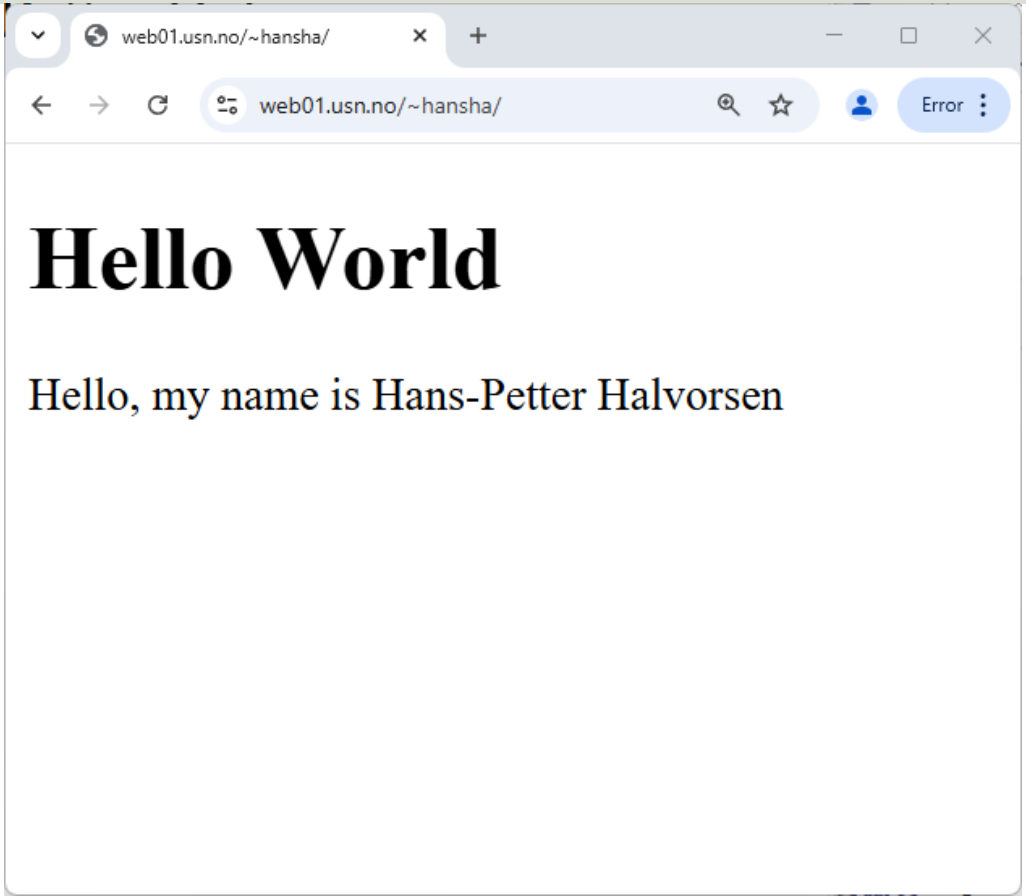
PHP Programming

Hans-Petter Halvorsen

Table of Contents

Hello World App





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 builtin support for Databases, i.e., inserting and retrieving data from different database systems.

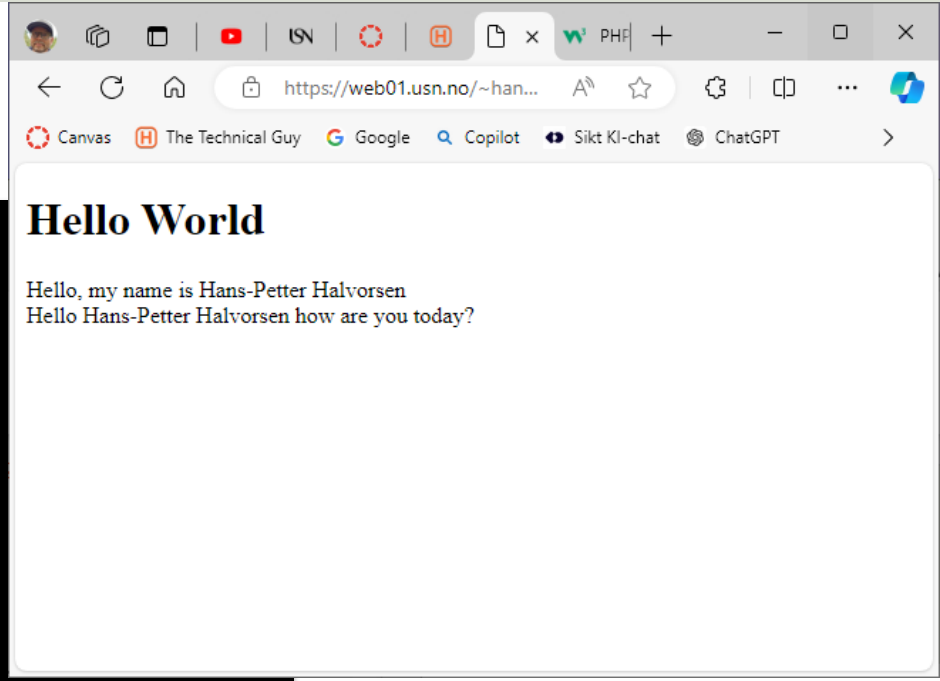
Here we will focus on the specific web features and database features, and not plain programming, since this is like C++/C# and any other programming language.

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

```php
<!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

C: > Users > hansha > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > HelloWorld > index.php

```php
<!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>
```

1

PHP code server-side

Final Result shown in the Web Browser

3

Hello World

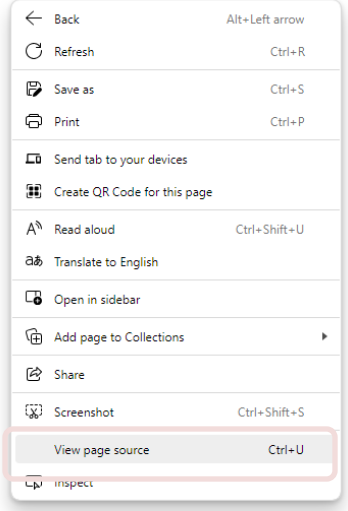
Hello, my name is Hans-Petter Halvorsen
Hello Hans-Petter Halvorsen how are you today?

Line wrap

2

```html
<!DOCTYPE html>
<html>
    <body>
        <h1>Hello World</h1>
        Hello, my name is Hans-Petter Halvorsen<br>Hello Hans-Petter Halvorsen how are you today?
    </body>
</html>
```

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

You will probably need to use many of these String function in your PHP Web Applications daily.

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 o 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.
- The syntax is very similar to C/C#.
- PHP supports many different types
- Here you see a basic example:

```
<!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

PHP Tutorial

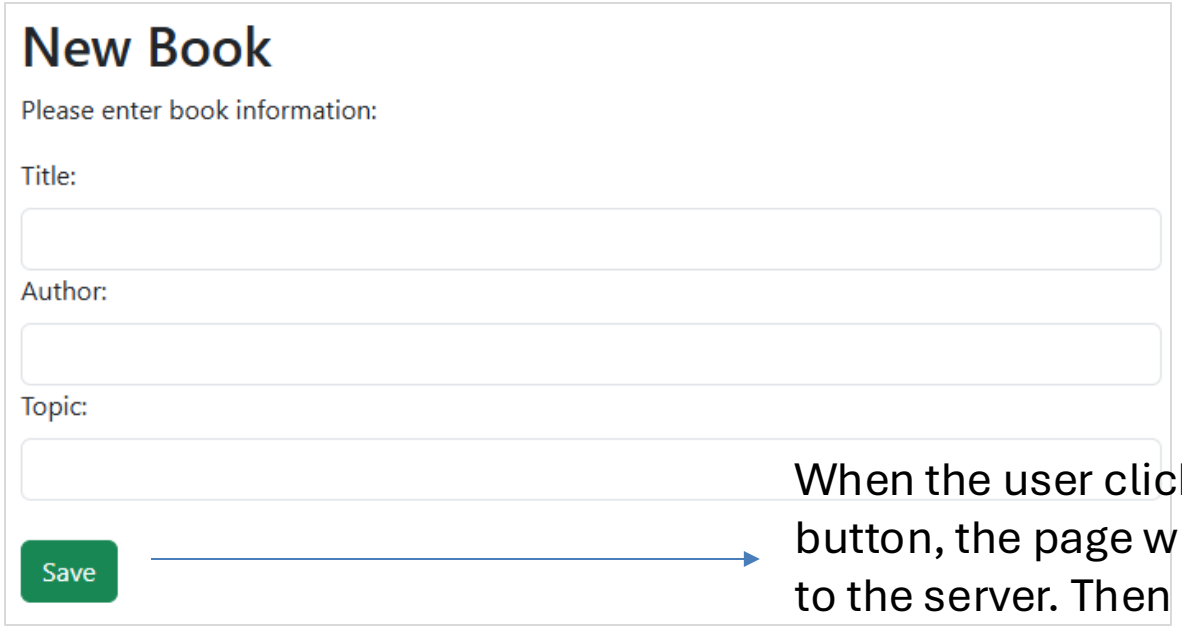
HTML Forms in PHP

Hans-Petter Halvorsen

Table of Contents

HTML Forms

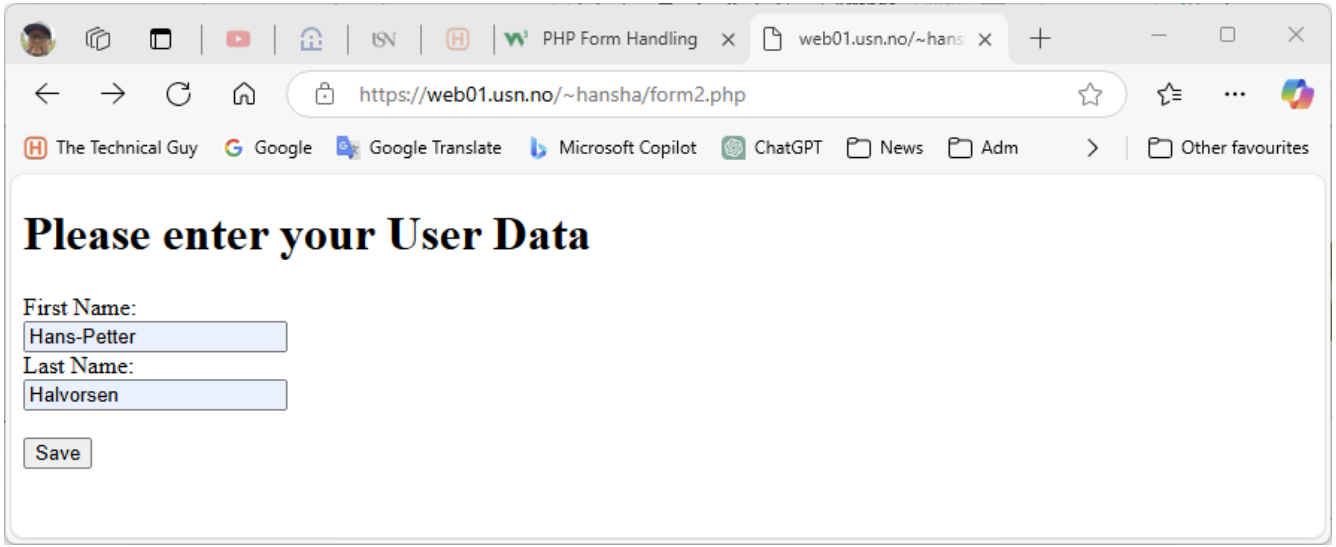
Example of a Web Page that contains an HTML Form:



When the user clicks on the Save button, the page will be sent back to the server. Then it is possible to tore the data in a database, etc.

HTML Forms

We start by creating a basic HTML Form like this:



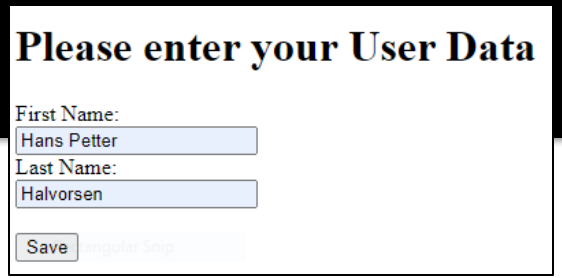
HTML Forms

- 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, e.g., in a Database like MySQL.

```
<!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

```html
<!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>
```




```php
<!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

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

GET Example

We can also $_GET

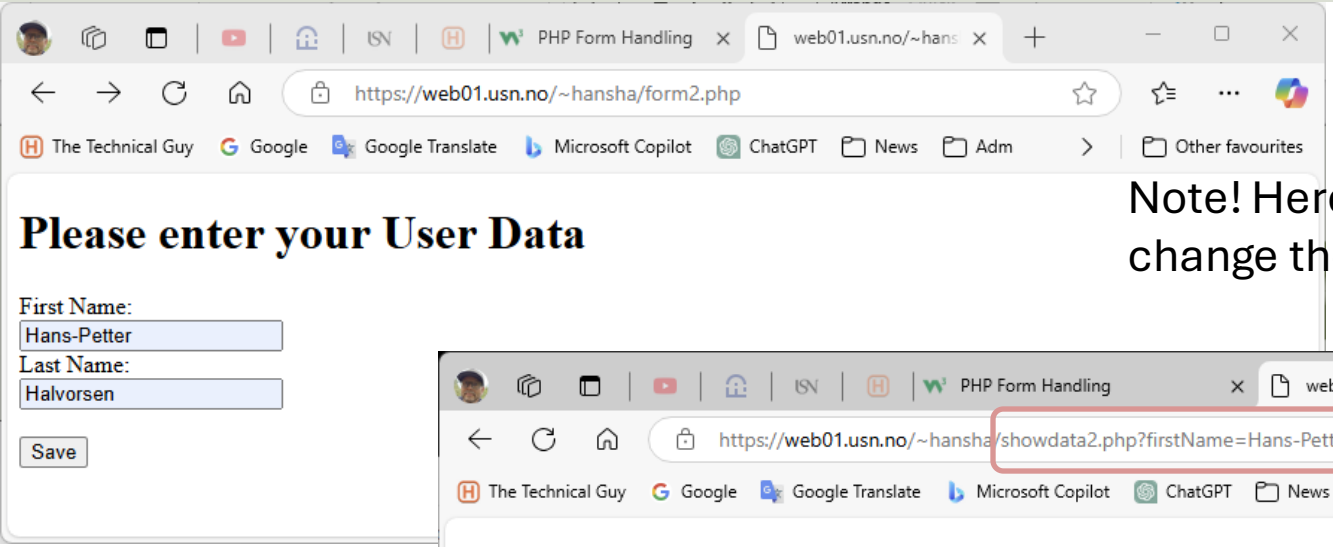
form2.php

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

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

<form action="showdata2.php" method="GET">
    <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>
```

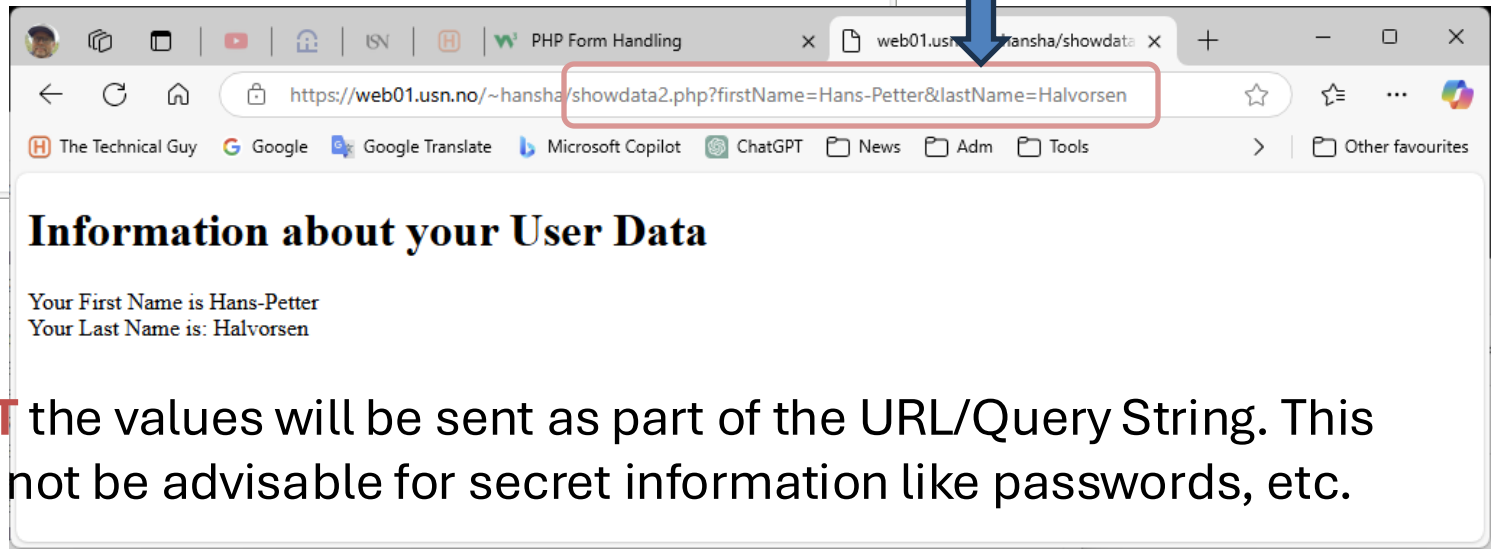
showdata2.php

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

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

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

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

</body>
</html>
```

GET and Query String



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



When using **GET** the values will be sent as part of the URL/Query String. This alternative may not be advisable for secret information like passwords, etc.

https://www.halvorsen.blog

PHP Tutorial

Session Variables in PHP

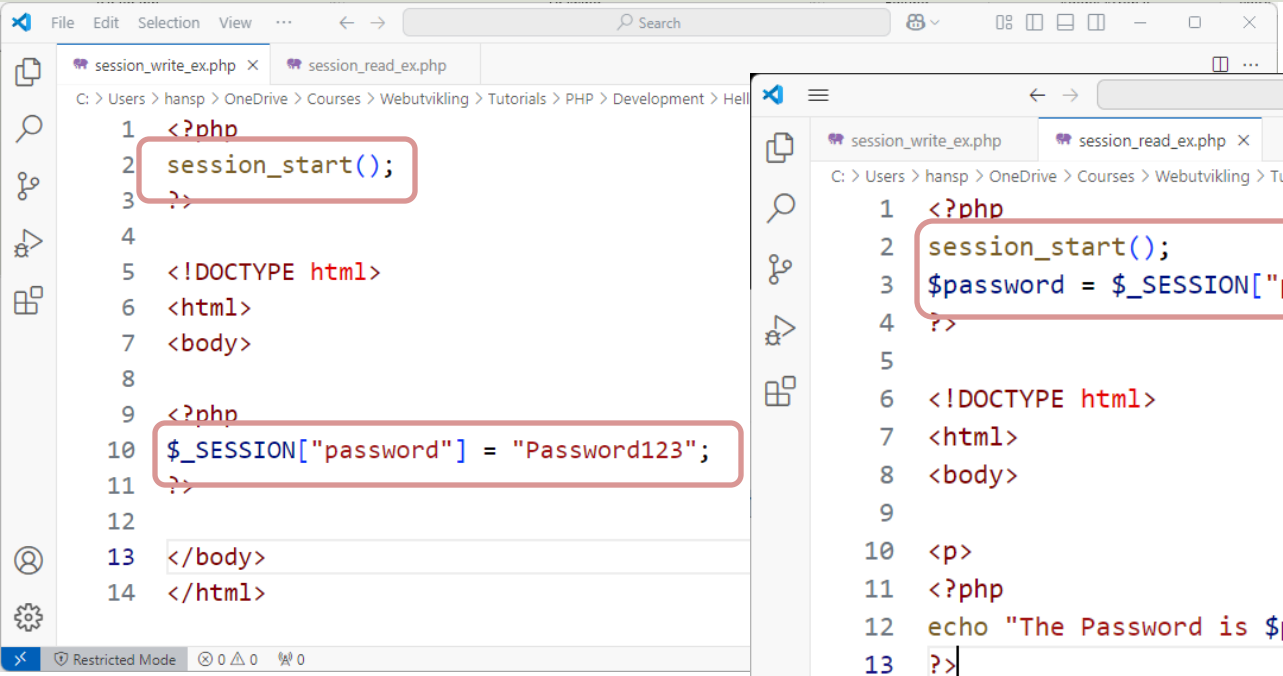
Hans-Petter Halvorsen

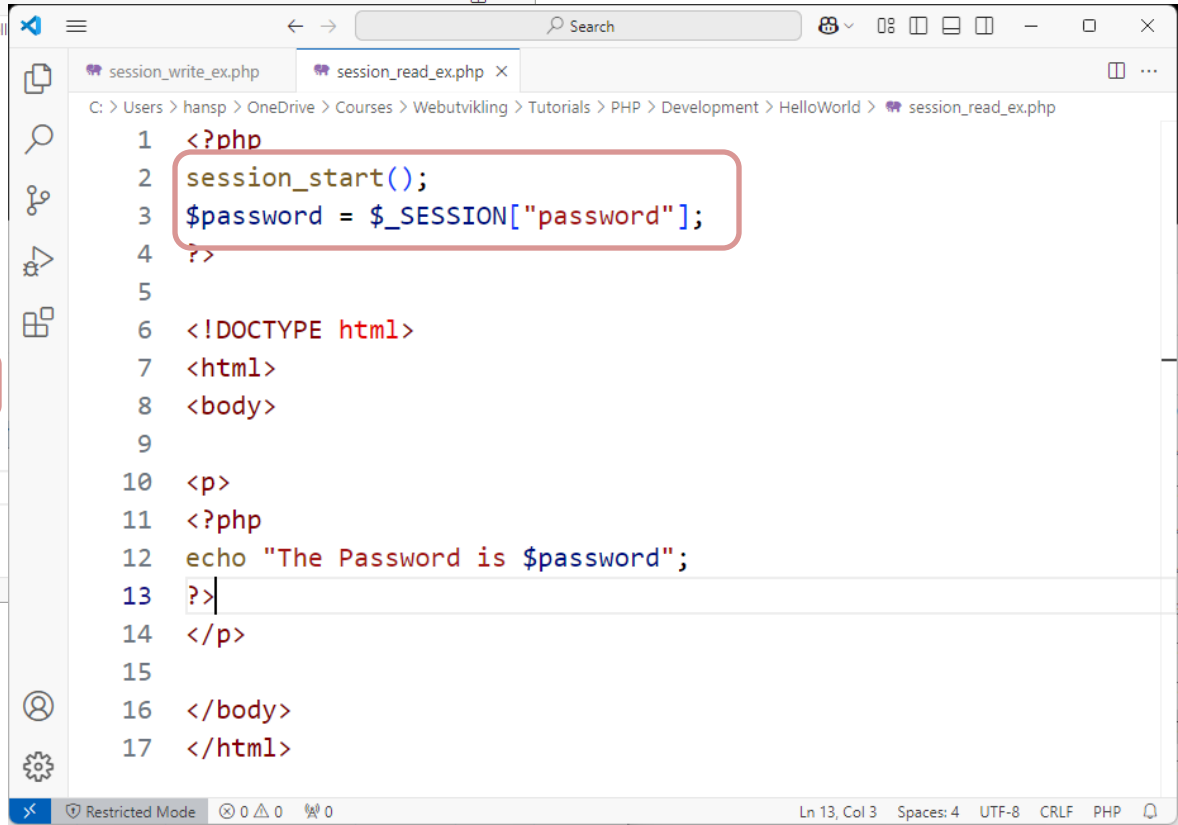
Table of Contents

Session Variables

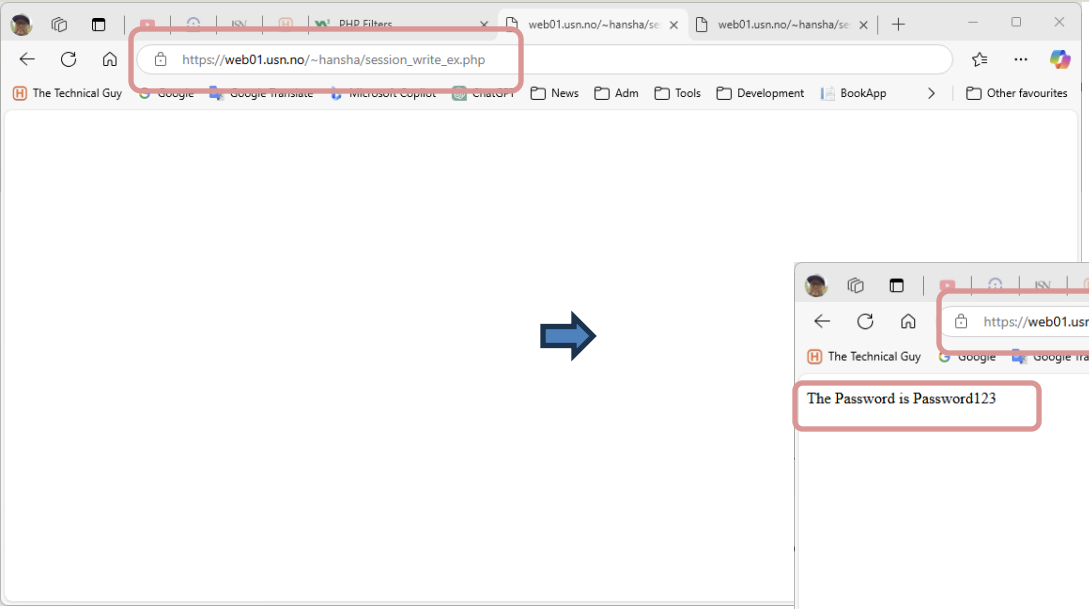
- We can use something called Session variables in order to send data between 2 web pages.
- Unlike a cookie, the information is not stored on the users PC.
- Session variables are very handy in web development.
- Session variables hold information only for one single user (you), so this means the information is only available for you and not for other users of the web page or web application.
- Sessions has a time limitation, typically 20 minutes.
- All web frameworks have these Sessions variables.

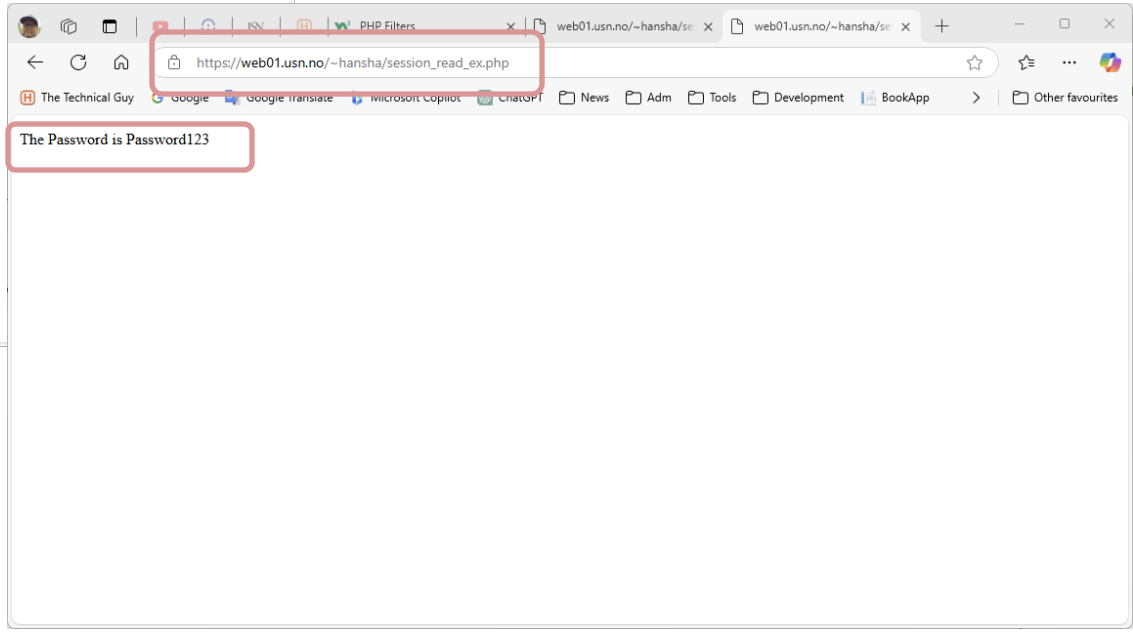
Session Example



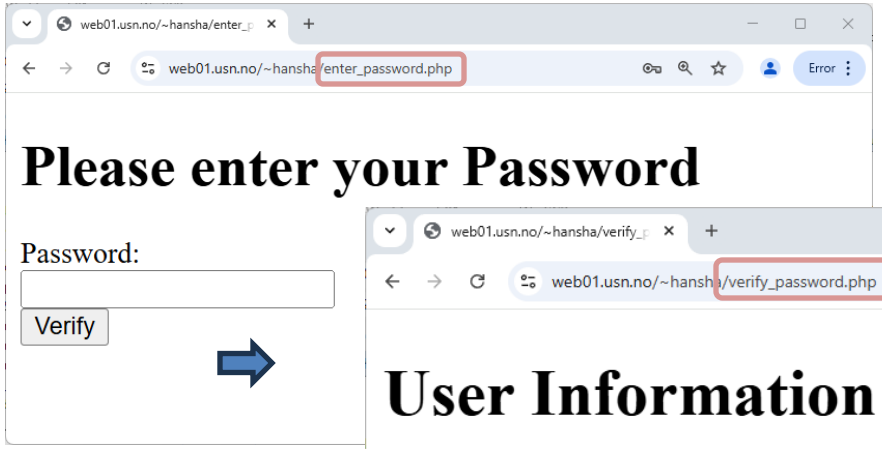


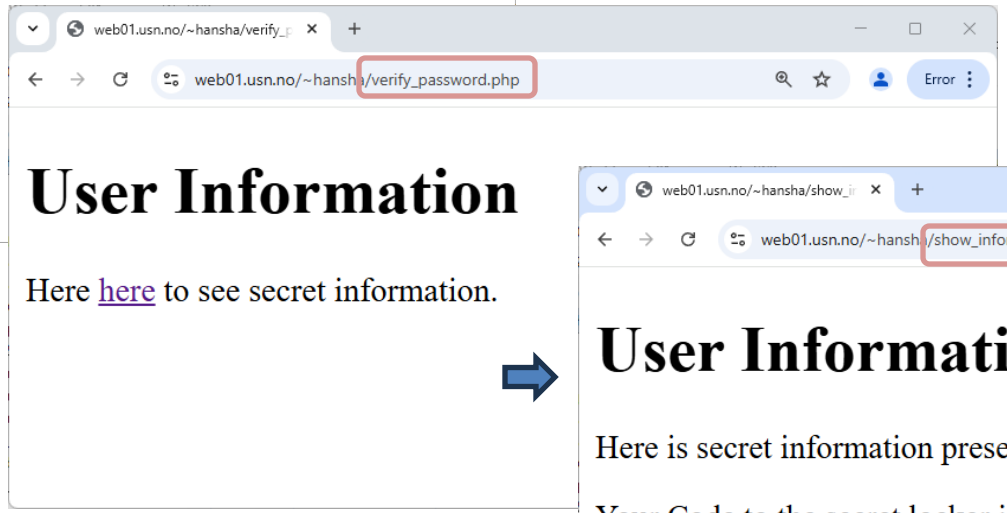
Session Example

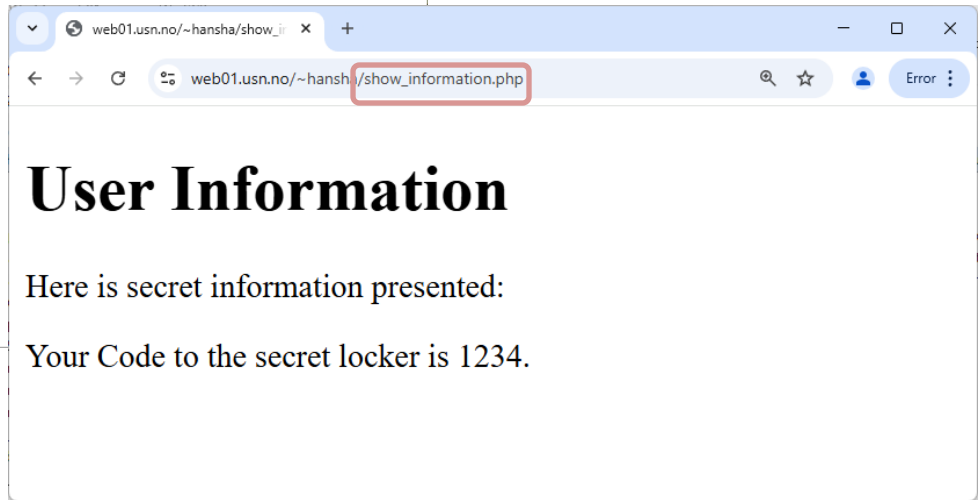


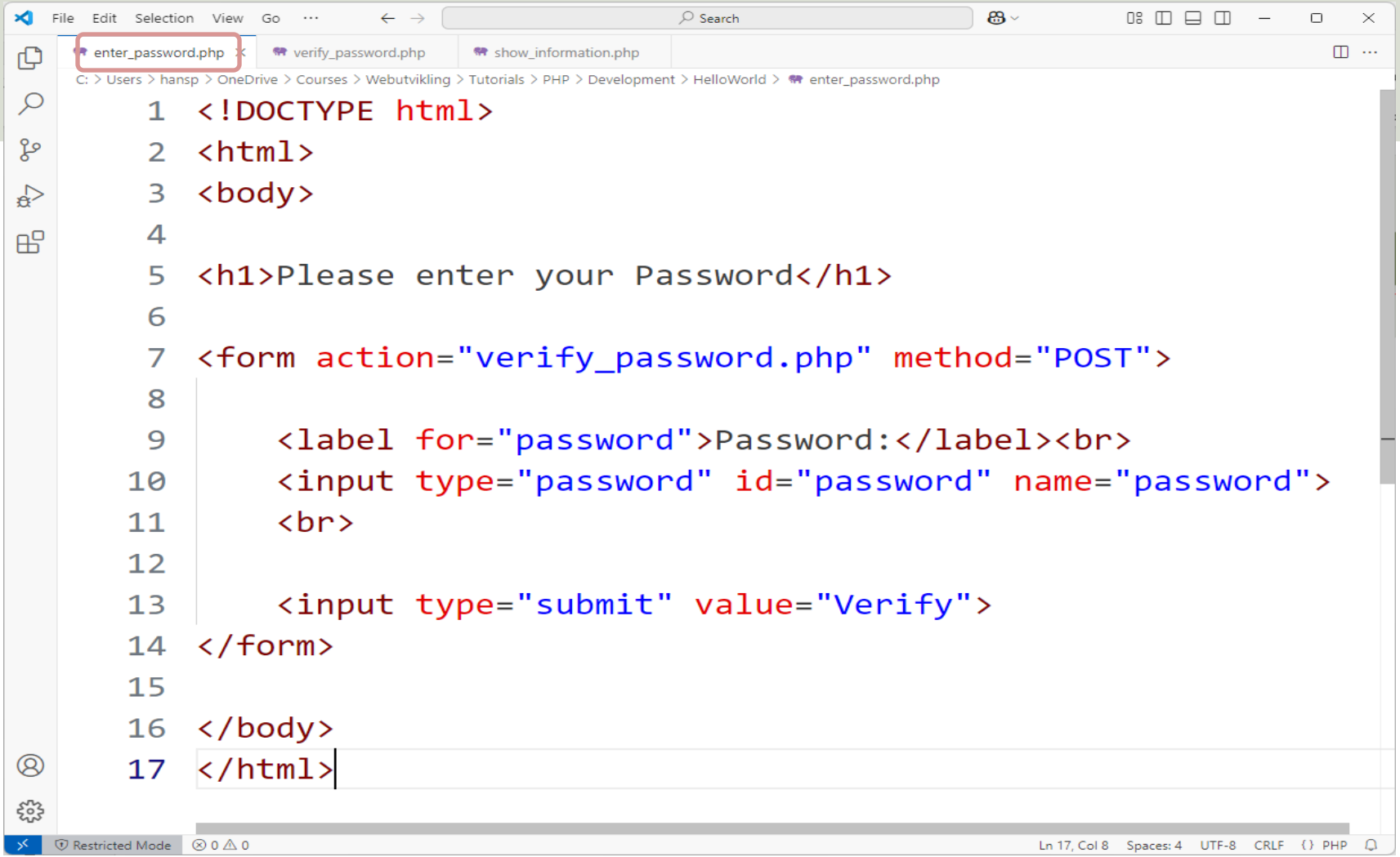


Session Example #2








File Edit Selection View Go
Search
enter_password.php
verify_password.php
show_information.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > HelloWorld > enter_password.php
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Please enter your Password</h1>
6
7 <form action="verify_password.php" method="POST">
8
9 <label for="password">Password:</label>
10 <input type="password" id="password" name="password">
11
12
13 <input type="submit" value="Verify">
14 </form>
15
16 </body>
17 </html>
Restricted Mode
Ln 17, Col 8 Spaces: 4 UTF-8 CRLF PHP

File Edit Selection View Go Run Terminal Help

Search

enter_password.php | verify_password.php | show_information.php

C: > Users > hansp > OneDrive > Desktop > Web Scripting Tutorials > PHP > Development > HelloWorld > verify_password.php

```php
<?php
session_start();
$password = $_POST["password"];

$authentication = false;
$_SESSION["authentication"] = false;
if ($password == "Password123")
{
    $authentication = true;
    $_SESSION["authentication"] = true;
}
?>
<!DOCTYPE html>
<html>
<body>
<h1>User Information</h1>

<?php
if ($authentication == true)
{
?>
<p> Here <a href="show_information.php">here</a> to see secret information.</p>
<?php
}
else
  echo "No information is available.";
?>

</body>
</html>
```

Restricted Mode | 0 0 | Ln 6, Col 22 | Spaces: 4 | UTF-8 | CRLF | PHP

File Edit Selection View ··· Search

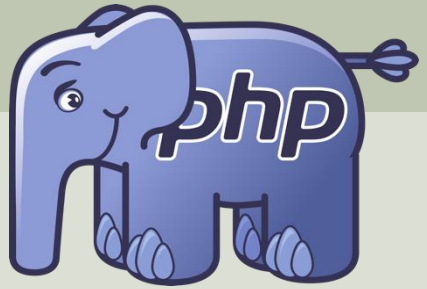
enter_password.php | verify_password.php | show_information.php

C: > Users > hansp > OneDrive > Courses > Webutvikling > tutorials > PHP > Development > HelloWorld > show_information.php

```php
<?php
session_start();
$authentication = $_SESSION["authentication"];
?>

<!DOCTYPE html>
<html>
<body>

<h1>User Information</h1>

<p> Here is secret information presented:</p>

<?php
if ($authentication == true)
{
?>
<p>
    Your Code to the secret locker is 1234.
<?php
}
?>

</body>
</html>
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 2, Col 17 Spaces: 4 UTF-8 CRLF { } PHP

https://www.halvorsen.blog

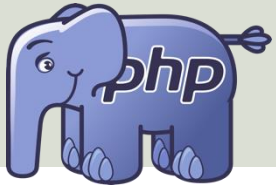


PHP Tutorial

PHP and MySQL Database

Hans-Petter Halvorsen

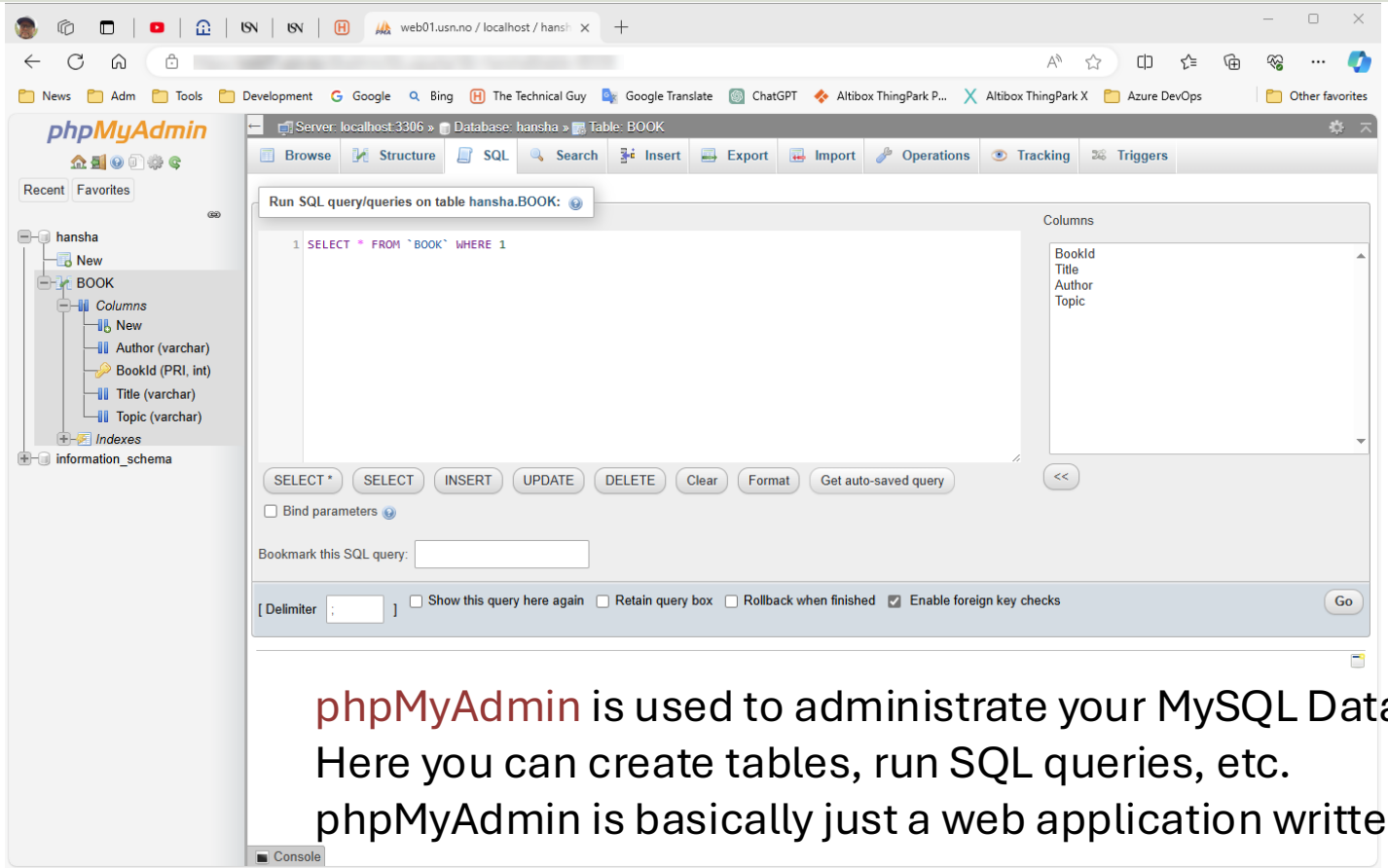
Table of Contents



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 and powerful.
- You can create powerful web applications where you can show, save, update and delete data in a MySQL database from the PHP code.

phpMyAdmin



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 to Database

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

```php
<?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:

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

CRUD

Typically, we want do the following operations:

- **C**reate (Insert) Data
- **R**ead (Select) Data
- **U**pdate Data
- **D**elete Data

=> This is referred to as CRUD functionality.

Typically, all Applications today need to communicate with a Database and have 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 “phpMyAdmin” or/and inside your PHP Web Application.
- SQL Example: select * from SCHOOL

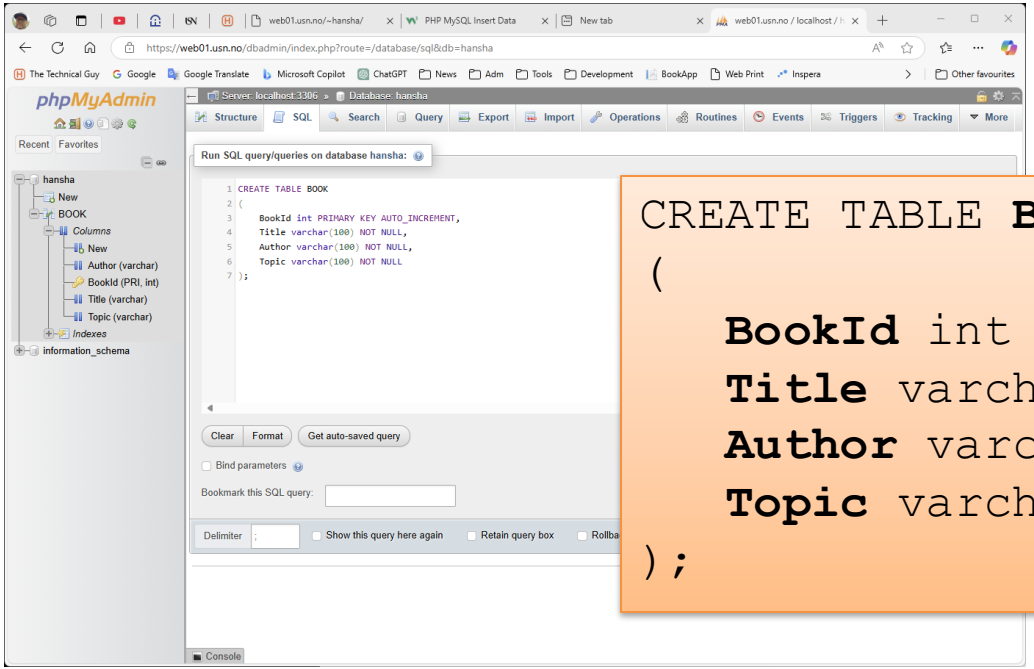
SQL Query Examples

```
• 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');
```

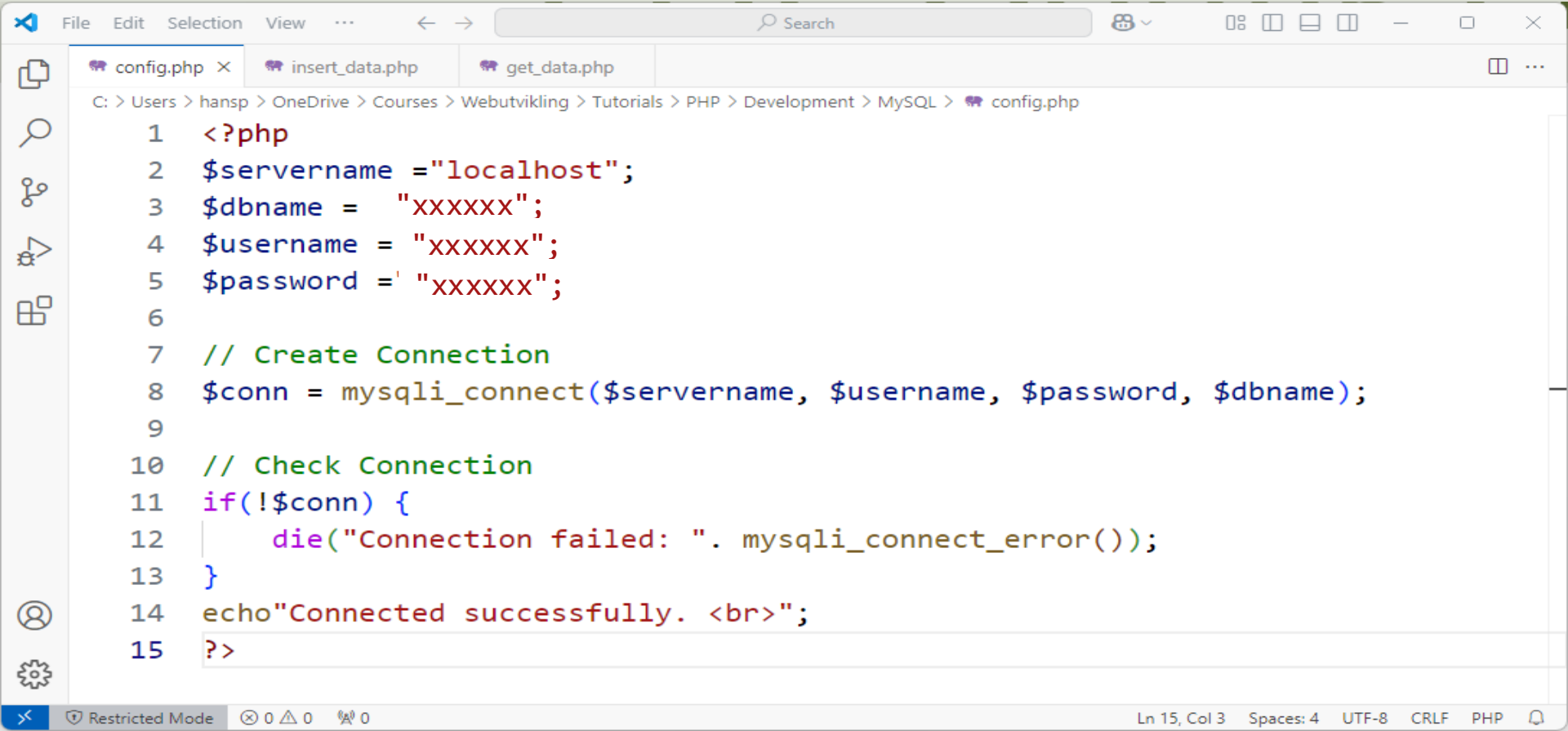
PHP Config File

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”. Then in the different PHP files we can include this file. This file will contain username, password, etc. for the MySQL Server database.

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

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

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

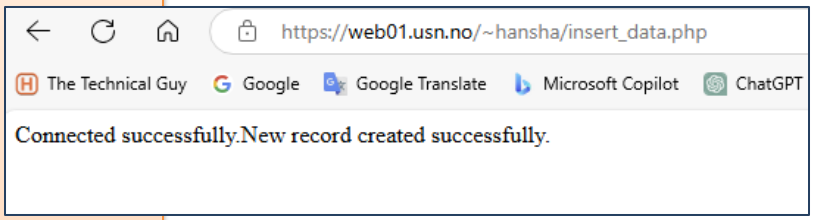
config.php

PHP Config File



Save Data to the Database

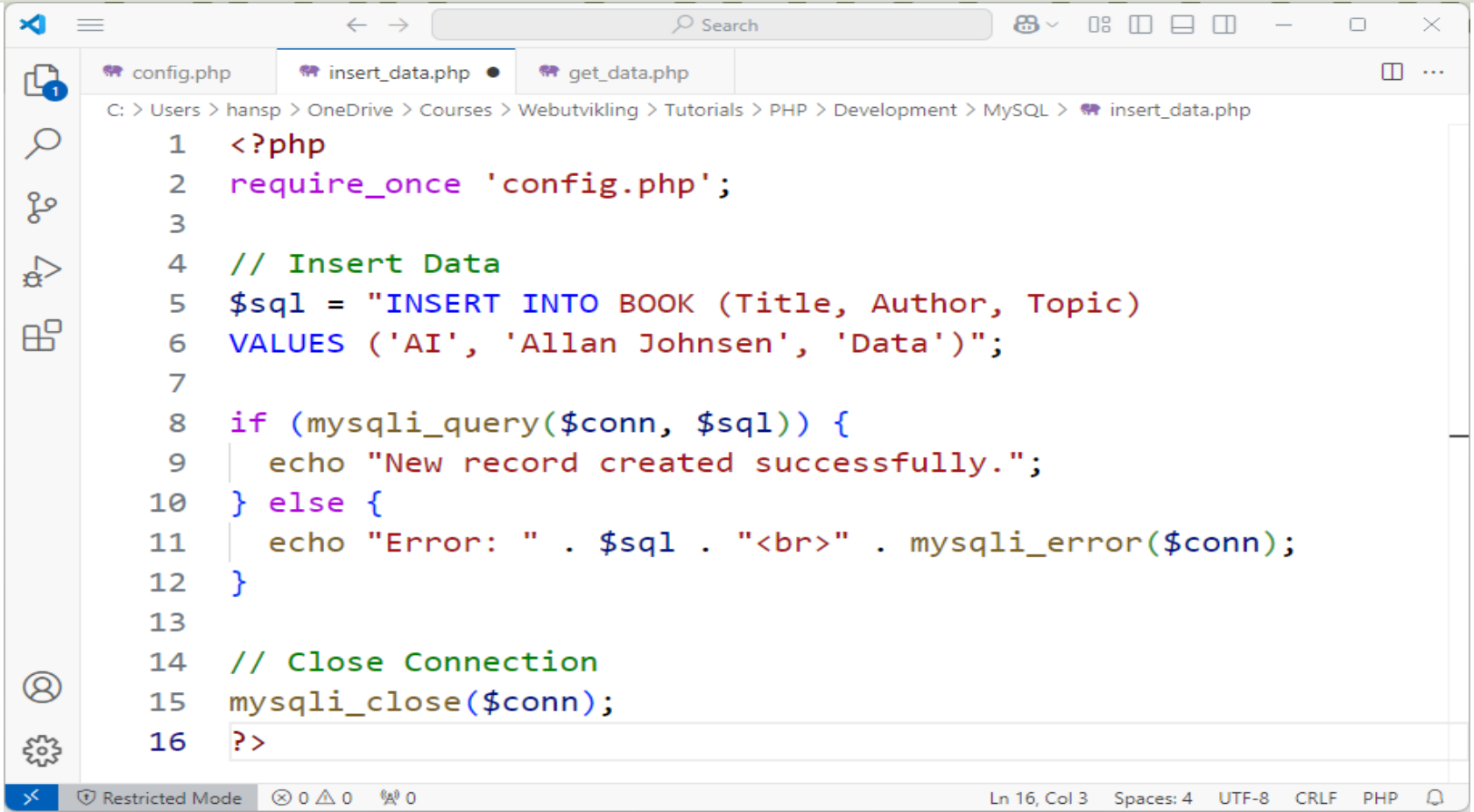
```php
<?php
require_once 'config.php';
// Insert Data
$sql = "INSERT INTO BOOK (Title, Author, Topic)
VALUES ('AI', 'Allan Johnsen', 'Data')";

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



Normally you get these Data from an HTML Form where the user enters this information.

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

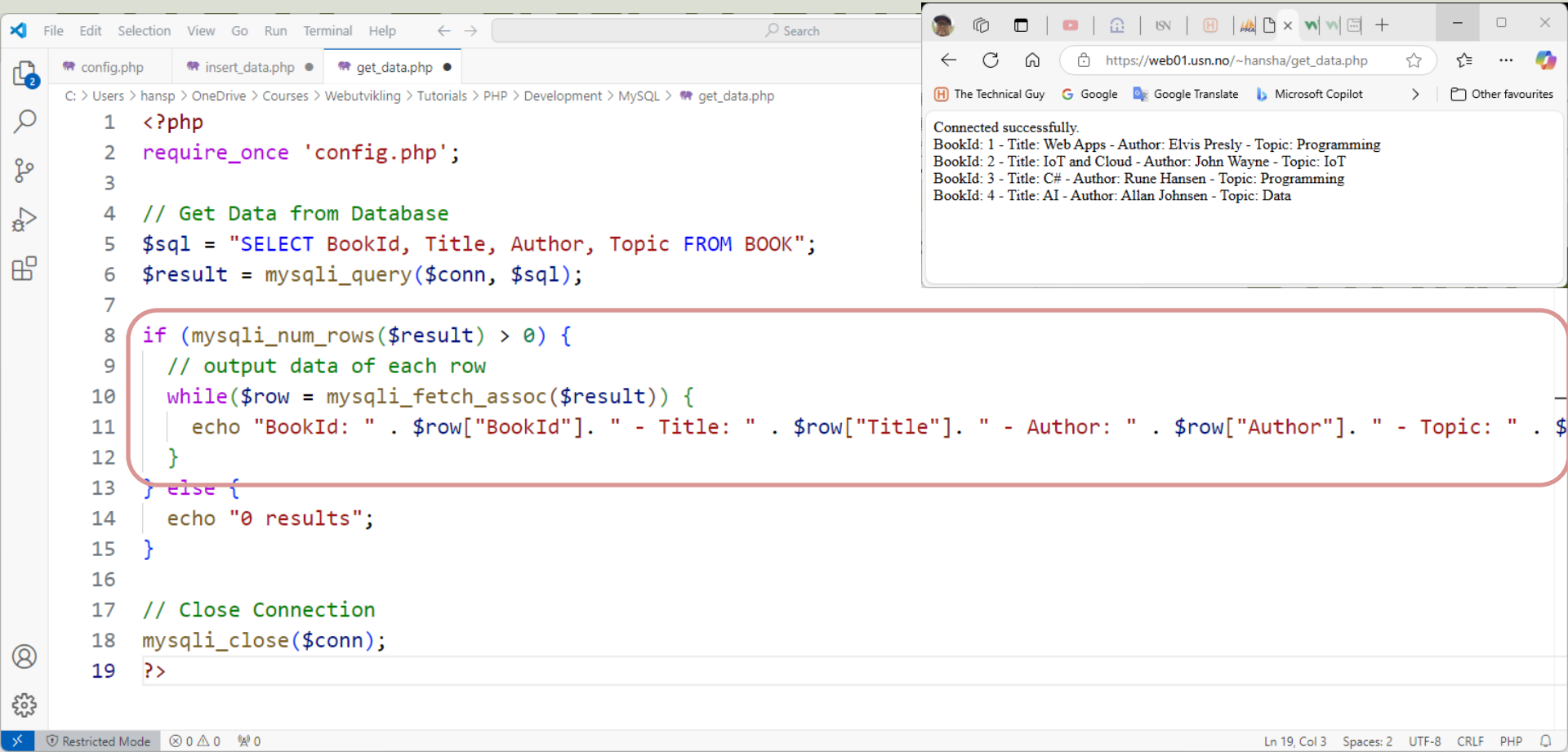
Save Data to the Database



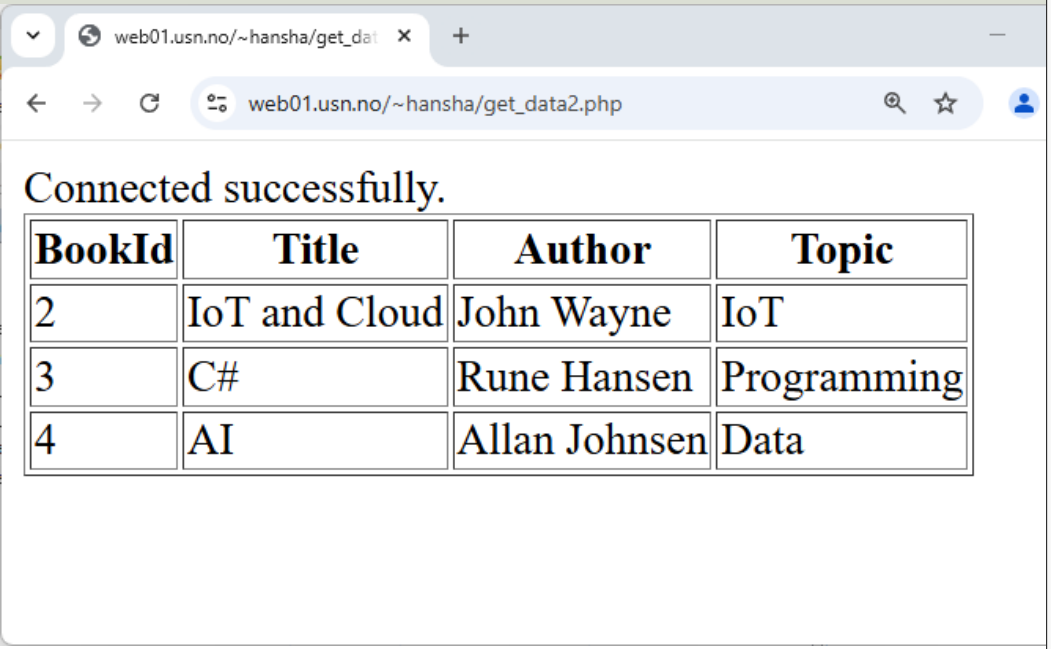
Show Data from the Database

```php
<?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



HTML Table




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

<!DOCTYPE html>
<html>

<body>

<table border="1">
  <thead>
    <tr>
      <th>BookId</th>
      <th>Title</th>
      <th>Author</th>
      <th>Topic</th>
    </tr>
  </thead>

  <tbody>

  <?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 "<tr>";
      echo "<td>" . $row["BookId"] . "</td>";
      echo "<td>" . $row["Title"] . "</td>";
      echo "<td>" . $row["Author"] . "</td>";
      echo "<td>" . $row["Topic"] . "</td>";
      echo "</tr>";
    }
  } else {
    echo "0 results";
  }
  ?>

  </tbody>
</table>

</body>
</html>

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

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