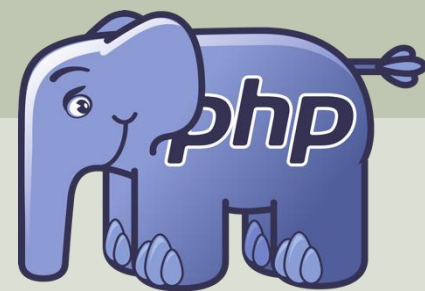


<https://www.halvorsen.blog>



# PHP and MySQL CRUD Web Application

Step by Step Instructions



Hans-Petter Halvorsen



# Contents

- [Introduction](#)
- [Database](#)
- [Connect to Database from PHP](#)
- [Read Data](#)
  - [Bootstrap](#)
- [Create Data](#)
- [Update Data](#)
- [Delete Data](#)
- [Summary and Further Work](#)

<https://www.halvorsen.blog>

# Introduction



[Table of Contents](#)

Hans-Petter Halvorsen

# Introduction

- We will create a **CRUD** Web Application using **PHP** and **MySQL**.
- CRUD means Create, Read, Update and Delete.
- This means our Web Application should retrieve data, insert data, update data and delete data from a MySQL database.
- We will go through and create the web application step by step by adding more and more functionality until finished.

# Introduction

- 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.

# Books Web Application

This the final PHP Web Application with CRUD functionality. But we will create it step by step doing one thing at the time.

## Books

Here you find a list of available books:

BookId	Title	Author	Topic	Action
1	Web Apps	Elvis Presly	Programming	<a href="#">Update Book</a> <a href="#">Delete Book</a>
2	IoT and Cloud	John Wayne	IoT	<a href="#">Update Book</a> <a href="#">Delete Book</a>
3	C#	Rune Hansen	Programming	<a href="#">Update Book</a> <a href="#">Delete Book</a>
4	AI	Allan Johnsen	Data	<a href="#">Update Book</a> <a href="#">Delete Book</a>

[New Book](#)

Do you really want to delete book #1'?

OK

Cancel

## New Book

Please enter book information:

Title:

Author:

Topic:

Save

## Update Book

Please enter book information:

Title:

Web Apps

Author:

Elvis Presly

Topic:

Programming

Save

# 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.
- 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>

# Database

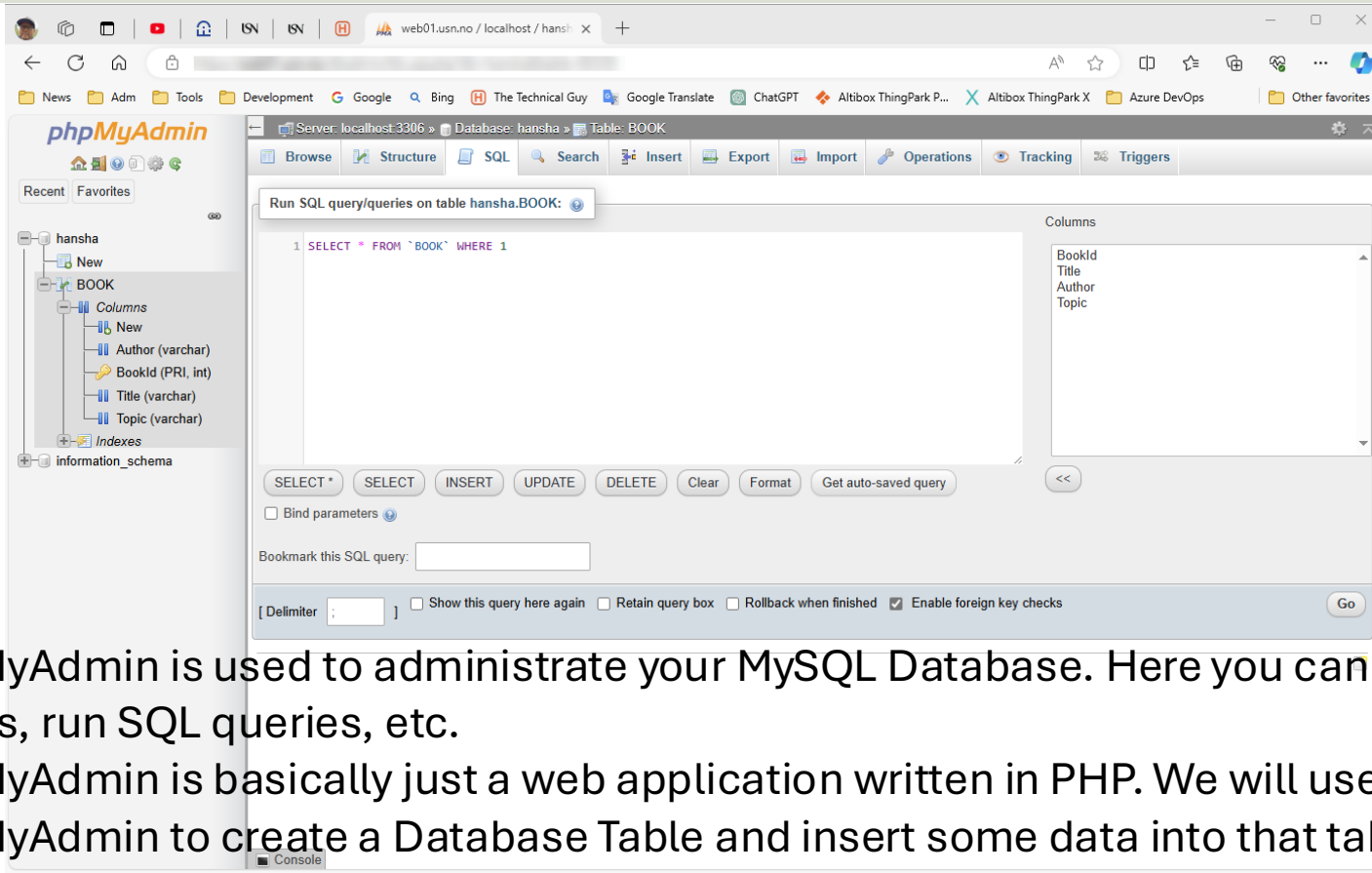


Hans-Petter Halvorsen

[Table of Contents](#)



# phpMyAdmin



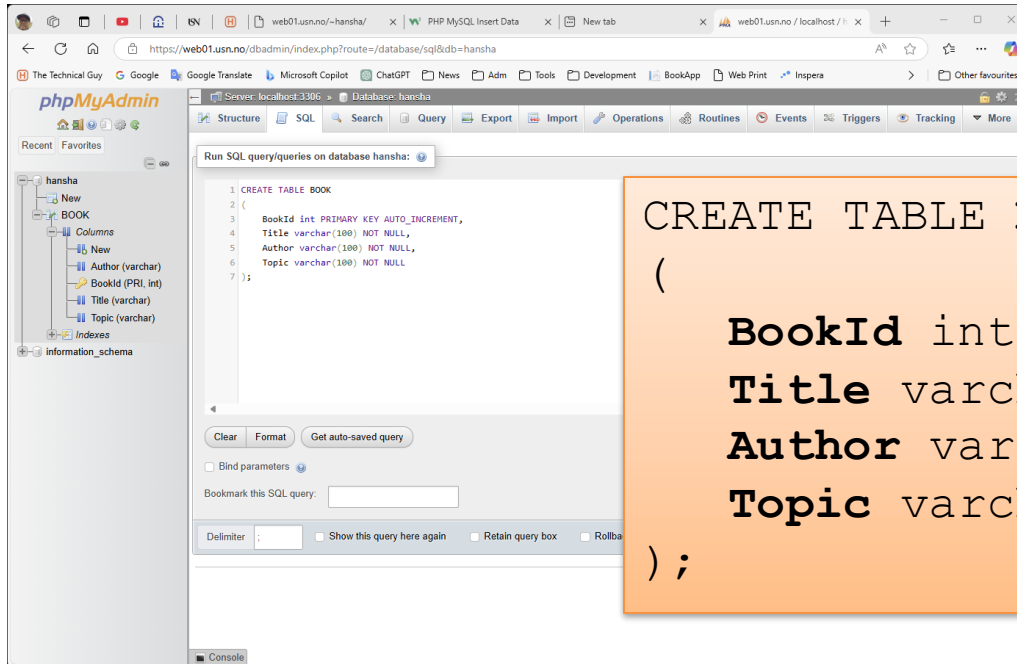
The screenshot displays the phpMyAdmin interface in a web browser. The browser's address bar shows the URL `web01.usn.no / localhost / hansha`. The phpMyAdmin interface includes a navigation sidebar on the left with a tree view showing the database structure: `hansha` (database), `New` (table), `BOOK` (table), `Columns` (columns), `Indexes` (indexes), and `information_schema` (database). The main content area is titled "Run SQL query/queries on table hansha.BOOK:". It features a toolbar with buttons for "Browse", "Structure", "SQL", "Search", "Insert", "Export", "Import", "Operations", "Tracking", and "Triggers". The "SQL" button is active, and the SQL query editor contains the text: `1 SELECT * FROM `BOOK` WHERE 1`. To the right of the editor is a "Columns" list showing `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 "Delimitter" (set to `;`), "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 interface.

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.



```
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 using phpMyAdmin, 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');
```

<https://www.halvorsen.blog>

# Connect to Database from PHP



Hans-Petter Halvorsen

[Table of Contents](#)

# 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

# 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.

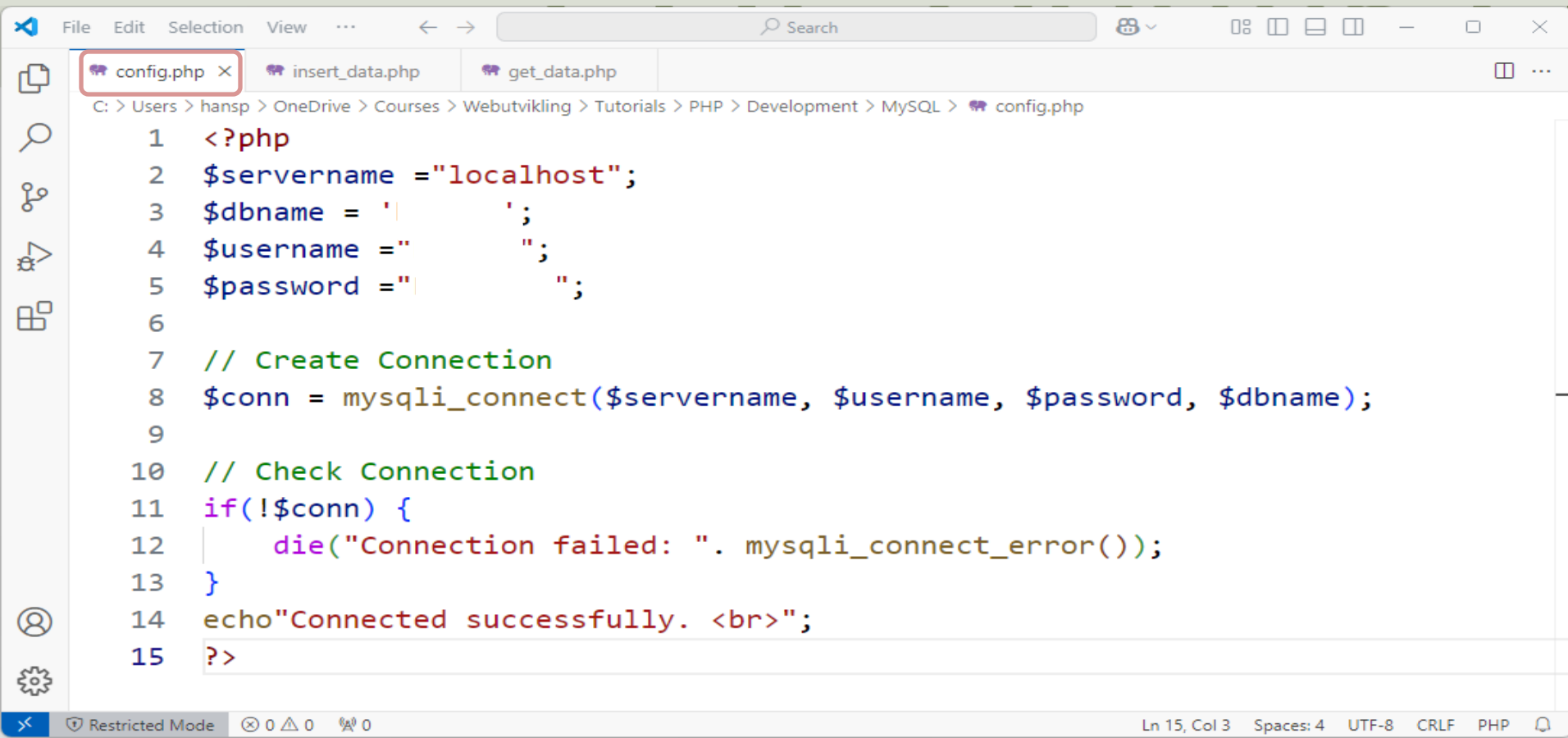
```
<?php
$servername = "localhost";
$username = "xxxxx";
$password = "xxxxx";
$dbname = "xxxxx";

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

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

config.php

# PHP Config File



```
1 <?php
2 $servername = "localhost";
3 $dbname = '      ';
4 $username = "      ";
5 $password = "      ";
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 ?>
```

Ln 15, Col 3 Spaces: 4 UTF-8 CRLF PHP

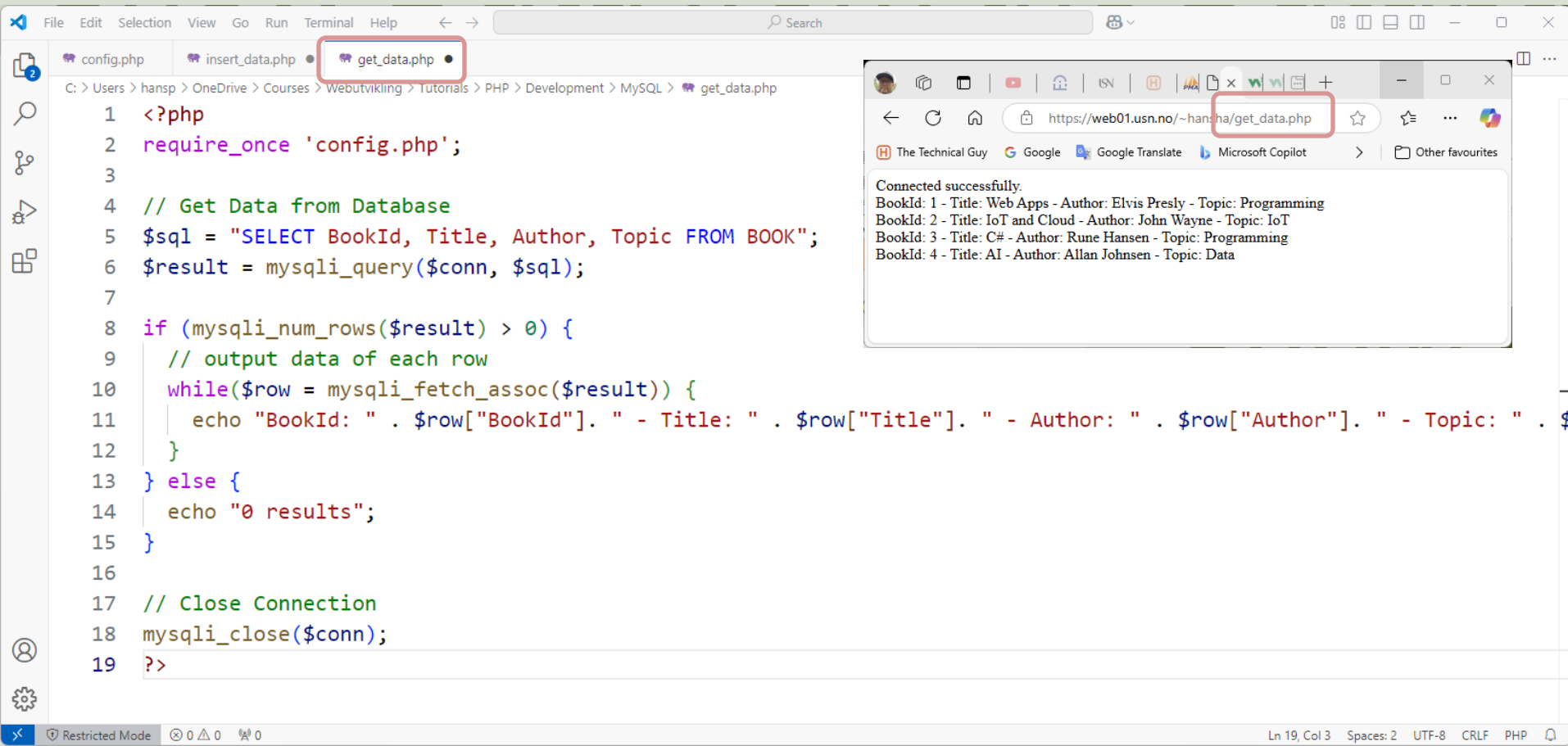
CRUD

# Read Data



Hans-Petter Halvorsen

# Show Data from the Database



The image shows a code editor window with a PHP script named `get_data.php` and a web browser window displaying the output of the script. The code editor window has a menu bar (File, Edit, Selection, View, Go, Run, Terminal, Help) and a search bar. The file explorer shows the path `C:\Users\hansp\OneDrive\Courses\Webutvikling\Tutorials\PHP\Development\MySQL\get_data.php`. The code in the editor is as follows:

```
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 ?>
```

The web browser window shows the URL `https://web01.usn.no/~hansha/get_data.php` and the output of the script:

```
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
```

The status bar at the bottom of the code editor shows "Ln 19, Col 3 Spaces: 2 UTF-8 CRLF PHP".

# Code

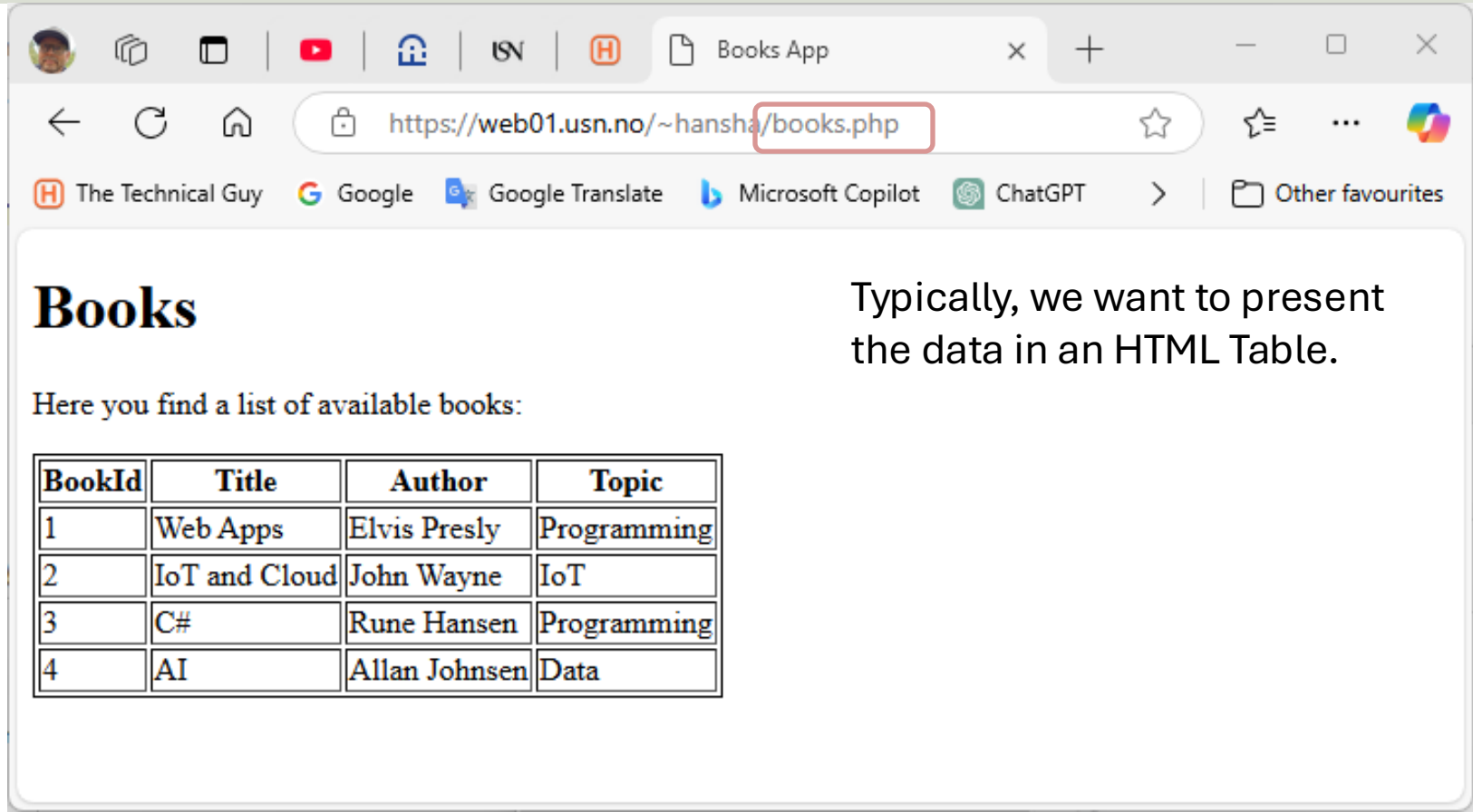
```
<?php
require_once 'config.php';

// 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);
?>
```

# Present Data in an HTML Table



The screenshot shows a web browser window with the address bar containing `https://web01.usn.no/~hansha/books.php`. The page title is "Books". Below the title, there is a heading "Books" and a paragraph: "Here you find a list of available books:". A table with 4 columns (BookId, Title, Author, Topic) and 4 rows of data is displayed. The table data is as follows:

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

Typically, we want to present the data in an HTML Table.



config.php

books.php X

C:\Users&gt; hansp&gt; OneDrive&gt; Courses&gt; Webutvikling&gt; Tutorials&gt; PHP&gt; Development&gt; CRUD Application&gt; Books CRUD Web Application&gt; books.php

```
18 <table border="1" style="border: 1px solid black;">
19 <thead>
20 <tr>
21 <th>BookId</th>
22 <th>Title</th>
23 <th>Author</th>
24 <th>Topic</th>
25 </tr>
26 </thead>
27
28 <tbody>
29 <?php
30 // Get Data from Database
31 $sql = "SELECT BookId, Title, Author, Topic FROM BOOK";
32 $result = mysqli_query($conn, $sql);
33
34 if (mysqli_num_rows($result) > 0) {
35     // output data of each row
36     while($row = mysqli_fetch_assoc($result)) {
37         echo "<tr><td>" . $row["BookId"] . "</td><td>" . $row["Title"] . "</td><td>" . $row["Author"] . "</td><td>" . $row["Topic"] . "</td></tr>";
38     }
39 } else {
40     echo "0 results";
41 }
42 ?>
43
44 </tbody>
45 </table>
```



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

<!DOCTYPE html>
<html>

<head>
<title>Books App</title>
</head>

<body>

<h1>Books</h1>

<p>Here you find a list of available books:</p>

<table border="1" style="border: 1px solid black;">
<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><td>" . $row["BookId"] . "</td><td>" . $row["Title"] . "</td><td>" . $row["Author"] . "</td><td>" . $row["Topic"] . "</td></tr>";
}
} else {
echo "0 results";
}
?>

</tbody>
</table>

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

</body>
</html>
```

<https://www.halvorsen.blog>

# Bootstrap



Hans-Petter Halvorsen

[Table of Contents](#)

# Bootstrap

- To create modern web pages and web applications today, it needs good design, layout and it needs to work on all kind of devices like PCs and smartphones, etc.
- Bootstrap is a popular CSS framework that is used by many web developers today. Bootstrap also uses JavaScript to create interactivity and animations, etc.
- By using such a framework like Bootstrap or similar you can focus on developing the application features and let Bootstrap do most of the layout, etc.
- <https://getbootstrap.com>

# Using Bootstrap

You can start using Bootstrap in different ways. The simplest method is to include the CSS and the JavaScript libraries in in the <head></head> section your HTML files.

## 1. Put Bootstrap CSS Library:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
```

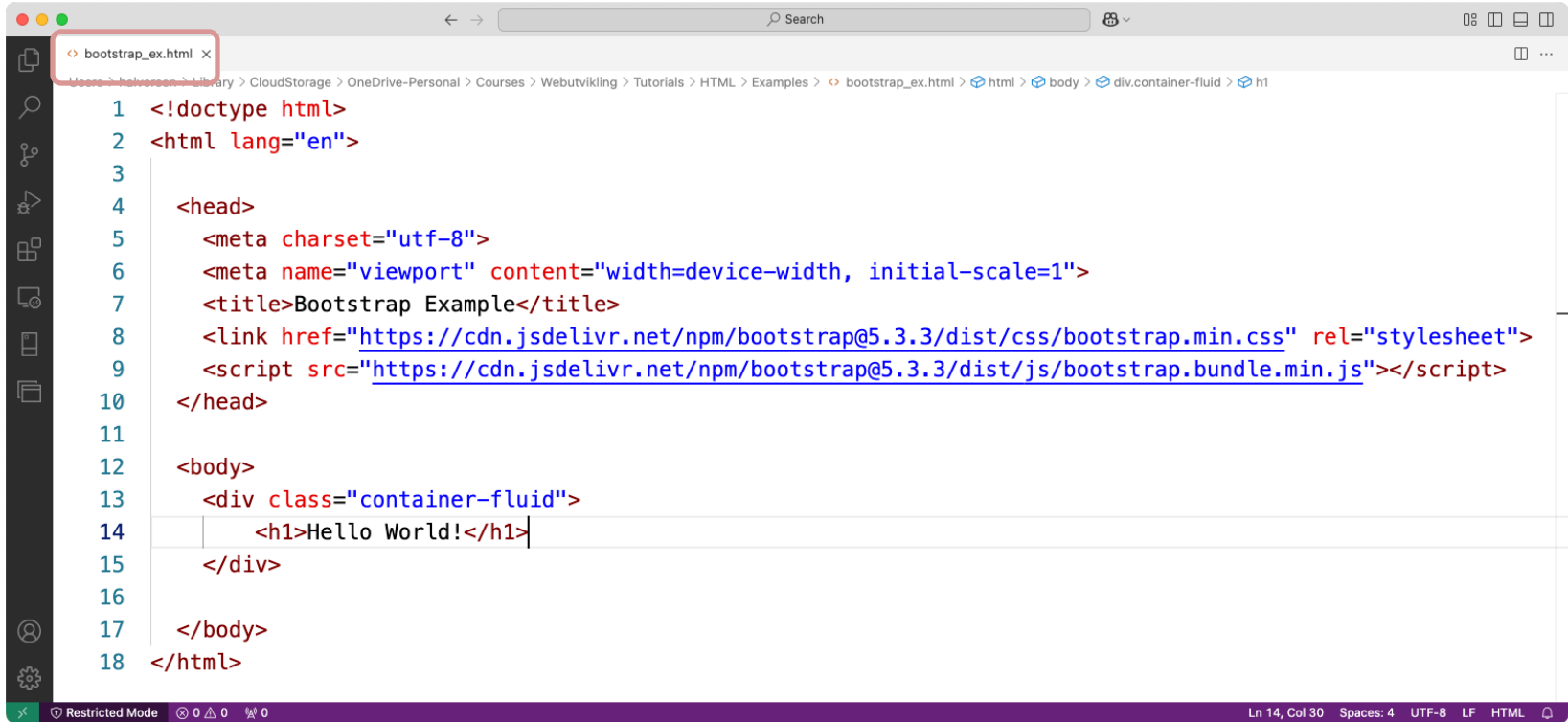
## 2. Bootstrap JavaScript Library :

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
```

3. This line of code ensures your code works fine on all devices like smartphones, etc:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

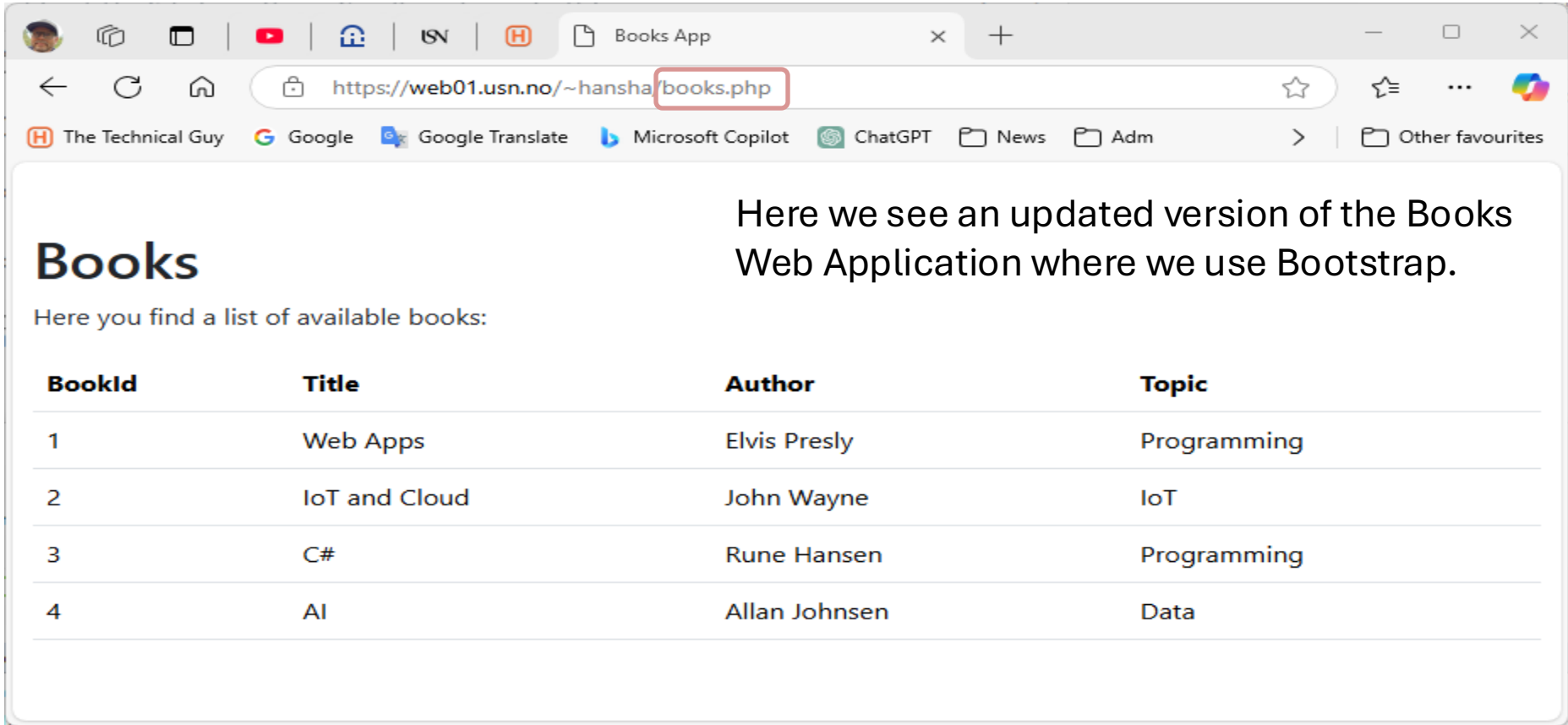
# Bootstrap - Hello World Example



```
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>Bootstrap Example</title>
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
9   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
10 </head>
11
12 <body>
13   <div class="container-fluid">
14     <h1>Hello World!</h1>
15   </div>
16
17 </body>
18 </html>
```

Ln 14, Col 30 Spaces: 4 UTF-8 LF HTML

# Updated Books Application



Books

Here you find a list of available books:

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

# Updated Code

“books.php”

```
books.php x
File Edit Selection View Go ...
C:\Users\> hoesp> C:\Drive > Courses > Webutvikling > Tutorials > PHP > Development > CRUD Application > Books CRUD Web Application > books.php
1 <?php
2 require_once 'config.php';
3 ?>
4
5 <!DOCTYPE html>
6 <html>
7
8 <head>
9 <title>Books App</title>
10 <meta charset="utf-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1">
12 <title>Bootstrap Example</title>
13 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
14 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
15 </head>
16
17 <body>
18 <div class="container-fluid pt-5">
19
20 <h1>Books</h1>
21
22 <p>Here you find a list of available books:</p>
23
24 <div class="table-responsive">
25 <table class="table">
26 <thead>
27 <tr>
28 <th>BookId</th>
29 <th>Title</th>
30 <th>Author</th>
31 <th>Topic</th>
32 </tr>
33 </thead>
34
35 <tbody>
36 <?php
37 // Get Data from Database
38 $sql = "SELECT BookId, Title, Author, Topic FROM BOOK";
39 $result = mysqli_query($conn, $sql);
40
41 if (mysqli_num_rows($result) > 0) {
42 // output data of each row
43 while($row = mysqli_fetch_assoc($result)) {
44 | echo "<tr><td>" . $row["BookId"] . "</td><td>" . $row["Title"] . "</td><td>" . $row["Author"] . "</td><td>" . $row["Topic"] . "</td></tr>";
45 | }
46 | } else {
47 | echo "0 results";
48 | }
49 | ?>
50
51 </tbody>
52 </table>
53 </div>
54
55 <?php
56 // Close Connection
57 mysqli_close($conn);
58 ?>
59
60 </div>
61 </body>
62 </html>
```

<https://www.halvorsen.blog>

CRUD

# Create Data



[Table of Contents](#)

Hans-Petter Halvorsen

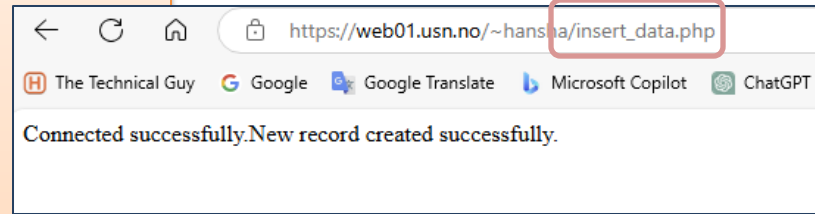


# Save Data to the Database

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

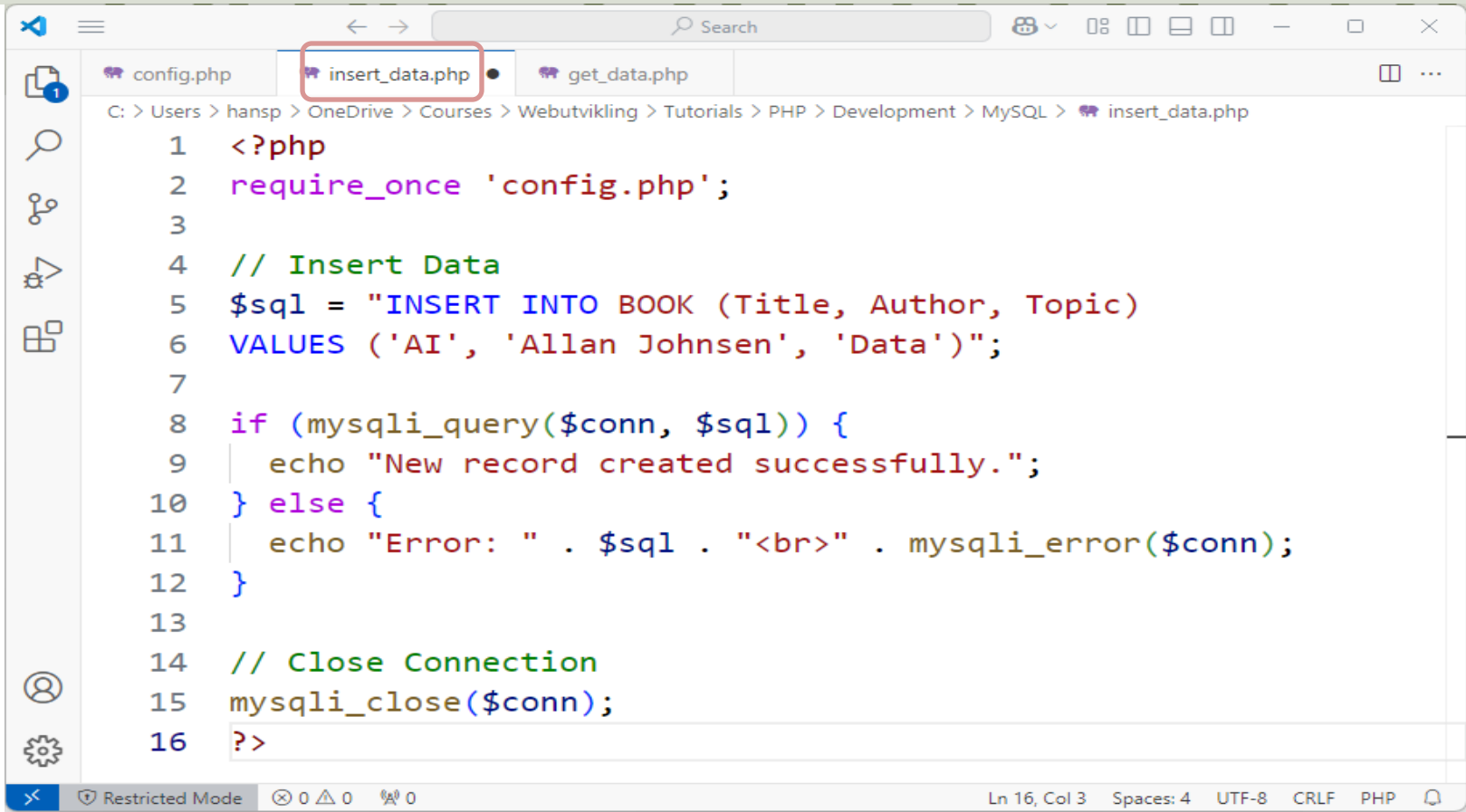
if (mysqli_query($conn, $sql)) {
    echo "New record created successfully.";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
// Close Connection
mysqli_close($conn);
?>
```

“insert\_data.php”



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

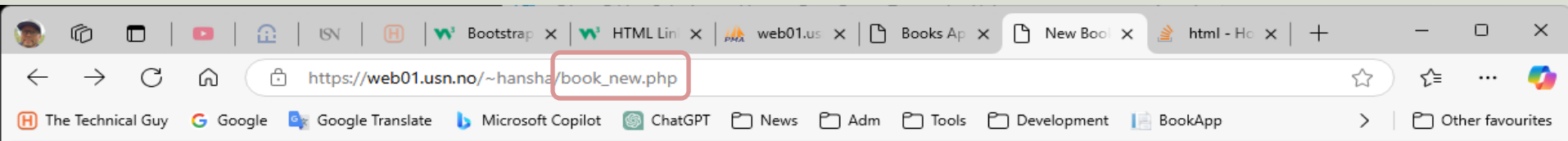
# Save Data to the Database



```
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 ?>
```

Ln 16, Col 3 Spaces: 4 UTF-8 CRLF PHP

# Create an HTML Form



A screenshot of a web browser window. The address bar shows the URL `https://web01.usn.no/~hansha/book_new.php`, which is highlighted with a red box. The browser's tab bar shows several open tabs, including 'Bootstrap', 'HTML Lin', 'web01.us', 'Books Ap', 'New Boo', and 'html - Ho'. The browser's toolbar includes navigation buttons (back, forward, refresh, home) and a search bar. Below the address bar, there are several search engines and services listed: 'The Technical Guy', 'Google', 'Google Translate', 'Microsoft Copilot', 'ChatGPT', 'News', 'Adm', 'Tools', 'Development', 'BookApp', and 'Other favourites'.

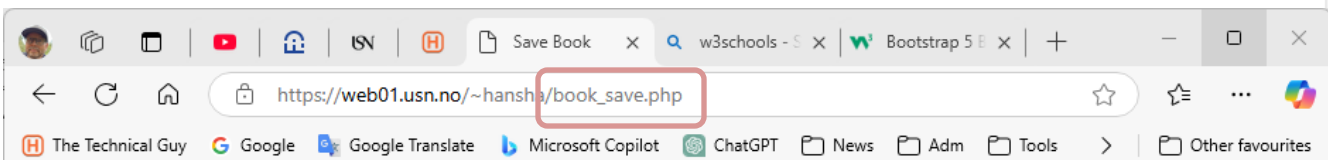
## New Book

Please enter book information:

Title:

Author:

Topic:



A screenshot of a web browser window. The address bar shows the URL `https://web01.usn.no/~hansha/book_save.php`, which is highlighted with a red box. The browser's tab bar shows several open tabs, including 'Save Book', 'w3schools - S', and 'Bootstrap 5 B'. The browser's toolbar includes navigation buttons (back, refresh, home) and a search bar. Below the address bar, there are several search engines and services listed: 'The Technical Guy', 'Google', 'Google Translate', 'Microsoft Copilot', 'ChatGPT', 'News', 'Adm', 'Tools', and 'Other favourites'.

## Save Book

New book created successfully.

```
File Edit Selection View Go Run ... Search
book_new.php x book_save.php
C:\Users\hansp...neDrive > Courses > Webutvikling > Tutorials > PHP > Development > CRUD Application > Books CRUD Web Application > book_new.php
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>New Book</title>
6   <meta charset="utf-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
9   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
10 </head>
11
12 <body>
13 <div class="container-fluid pt-5">
14
15 <h1>New Book</h1>
16 <p>Please enter book information:</p>
17
18 <div class="form-group">
19 <form action="book_save.php" method="POST">
20   <label for="title" class="form-label">Title:</label>
21   <input type="text" id="title" name="title" class="form-control">
22
23   <label for="author" class="form-label">Author:</label>
24   <input type="text" id="author" name="author" class="form-control">
25
26   <label for="topic" class="form-label">Topic:</label>
27   <input type="text" id="topic" name="topic" class="form-control">
28
29   <br>
30   <input type="submit" value="Save" class="btn btn-success">
31 </form>
32 </div>
33
34 </div>
35 </body>
36 </html>
```

“book\_new.php”

```
File Edit Selection View Go Run ... Search
book_save.php X
C:\Users> hansi> OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > CRUD Application > Books CRUD Web Application > book_save.php
1 <?php
2 require_once 'config.php';
3
4 //Get Post Data
5 $title = $_POST["title"];
6 $author = $_POST["author"];
7 $topic = $_POST["topic"];
8
9 // Insert Data
10 $sql = "INSERT INTO BOOK (Title, Author, Topic)
11 VALUES ('$title', '$author', '$topic')";
12
13 if (mysqli_query($conn, $sql)) {
14     $message = "New book created successfully.<br>";
15 }
16 else {
17     $message = "Error: " . $sql . "<br>" . mysqli_error($conn);
18 }
19
20 // Close Connection
21 mysqli_close($conn);
22 ?>
23
24 <!DOCTYPE html>
25 <html>
26
27 <head>
28 <title>Save Book</title>
29 <meta charset="utf-8">
30 <meta name="viewport" content="width=device-width, initial-scale=1">
31 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
32 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
33 </head>
34
35 <body>
36 <div class="container-fluid pt-5">
37
38 <h1>Save Book</h1>
39
40 <?php echo $message ?><br>
41
42 <a href="books.php" class="btn btn-success">Go back to Books</a>
43
44 </div>
45 </body>
46 </html>
```

“book\_save.php”

# Updated “books.php”

We add a button so we can go to the New Book page:

The image illustrates the workflow for adding a new book to a system. It consists of three sequential browser screenshots:

- Books Page:** A browser window showing a page titled "Books". Below the title, it says "Here you find a list of available books:". A table lists books with columns for BookId, Title, and Author. A green "New Book" button is highlighted with a red box at the bottom left. A blue arrow points from this button to the next screenshot.
- New Book Form:** A browser window showing a form titled "New Book". The form asks for "Please enter book information:" and includes three input fields: "Title:", "Author:", and "Topic:". A green "Save" button is at the bottom left. A blue arrow points from the "Save" button to the next screenshot.
- Save Book Confirmation:** A browser window showing a confirmation message titled "Save Book" with the text "New book created successfully.". A green "Go back to Books" button is at the bottom.

BookId	Title	Author
1	Web Apps	Elvis Presly
2	IoT and Cloud	John Wayne
3	C#	Rune Hansen
4	AI	Allan Johnsen
25	PHP Programming	Hans Hansen
26	DeepSeek	Allan Triumph

# Updated “books.php”

We add a button so we can go to the New Book page:

```
56 |  
57 |     </tbody>  
58 | </table>  
59 | </div>  
60 |  
61 | <a href="book_new.php" class="btn btn-success">New Book</a>  
62 |
```

<https://www.halvorsen.blog>

CRUD

# Update Data



Hans-Petter Halvorsen

[Table of Contents](#)



# Updated “books.php”

The screenshot shows a web browser window with the address bar containing `https://web01.usn.no/~hansha/books.php`. The page title is "Books". Below the title, it says "Here you find a list of available books:". A table lists books with columns for BookId, Title, Author, Topic, and Action. The Action column contains "Update Book" buttons for each row. A red rounded rectangle highlights the "Update Book" buttons. At the bottom left, there is a "New Book" button.

BookId	Title	Author	Topic	Action
1	Web Apps	Elvis Presly	Programming	Update Book
2	IoT and Cloud	John Wayne	IoT	Update Book
3	C#	Rune Hansen	Programming	Update Book
4	AI	Allan Johnsen	Data	Update Book
25	PHP Programming	Hans Hansen	Programming	Update Book
26	DeepSeek	Allan Triumph	AI	Update Book
36	DeepSeek	Allan Triumph	AI	Update Book

New Book

We add “Update Book” buttons so we can update a selected Book:

```
24 <table class="table">
25 <thead>
26 <tr>
27 <th>BookId</th>
28 <th>Title</th>
29 <th>Author</th>
30 <th>Topic</th>
31 <th>Action</th>
32 </tr>
33 </thead>
```

We add a “Update Book” buttons so we update a selected Book:

```
34 <tbody>
35 <?php
36 // Get Data from Database
37 $sql = "SELECT BookId, Title, Author, Topic FROM BOOK";
38 $result = mysqli_query($conn, $sql);
39
40
41 if (mysqli_num_rows($result) > 0) {
42 // output data of each row
43 while($row = mysqli_fetch_assoc($result)) {
44 echo "<tr>";
45 echo "<td>" . $row["BookId"] . "</td>";
46 echo "<td>" . $row["Title"] . "</td>";
47 echo "<td>" . $row["Author"] . "</td>";
48 echo "<td>" . $row["Topic"] . "</td>";
49 echo "<td> <a href='book_update.php?bookid=" . $row["BookId"] . "' class='btn btn-info'>Update Book</a> </td>";
50 echo "</tr>";
51 }
52 } else {
53 echo "0 results";
54 }
55 ?>
```

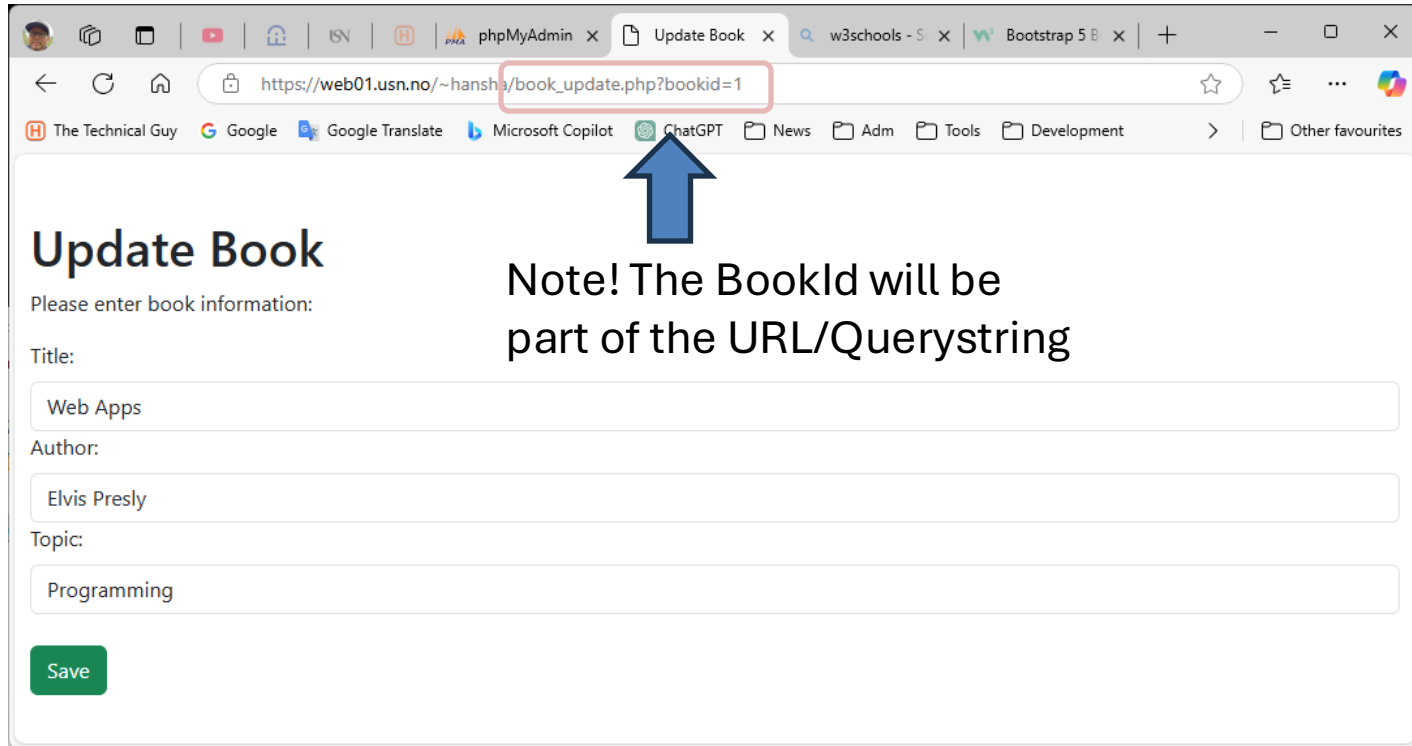
We add a Hyperlink for each row

Each Book has a unique BookId

```
56 </tbody>
57 </table>
```

# We create “book\_update.php”

We make a copy of the existing “book\_new.php” and makes the necessary changes:



The screenshot shows a web browser window with the URL `https://web01.usn.no/~hansha/book_update.php?bookid=1` highlighted in a red box. A blue arrow points from the text below to the `bookid=1` part of the URL. The browser's address bar also shows several open tabs: phpMyAdmin, Update Book, w3schools, and Bootstrap 5. The page content includes a form with the following fields:

- Title: Web Apps
- Author: Elvis Presly
- Topic: Programming
- Save button

**Note! The BookId will be part of the URL/Querystring**

```
File Edit Selection View Search
book_update.php x
C:\Users\hansp>OneDrive>Courses>Webutvikling>Tutorials>PHP>Development>CRUD Application>Books CRUD Web Application>book_update.php
1 <?php
2 require_once 'config.php';
3
4 //Get GET Data
5 $bookid = $_GET["bookid"];
6
7 // Get specific Book Data from Database
8 $sql = "SELECT Title, Author, Topic FROM BOOK WHERE BookId = $bookid";
9 $result = mysqli_query($conn, $sql);
10
11 if (mysqli_num_rows($result) > 0) {
12     $row = mysqli_fetch_assoc($result);
13     $title = $row["Title"];
14     $author = $row["Author"];
15     $topic = $row["Topic"];
16 } else {
17     echo "0 results";
18 }
19
20 ?>
21
22 <!DOCTYPE html>
23 <html>
24
25 <head>
26 <title>Update Book</title>
27 <meta charset="utf-8">
28 <meta name="viewport" content="width=device-width, initial-scale=1">
29 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
30 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
31 </head>
32
33 <body>
34 <div class="container-fluid pt-5">
35
36 <h1>Update Book</h1>
37 <p>Please enter book information:</p>
38
39 <div class="form-group">
40 <form action="book_update_db.php?bookid=<?php echo $bookid ?>" method="POST">
41 <label for="title" class="form-label">Title:</label>
42 <input type="text" id="title" name="title" class="form-control" value="<?php echo $title ?>">
43
44 <label for="author" class="form-label">Author:</label>
45 <input type="text" id="author" name="author" class="form-control" value="<?php echo $author ?>">
46
47 <label for="topic" class="form-label">Topic:</label>
48 <input type="text" id="topic" name="topic" class="form-control" value="<?php echo $topic ?>">
49
50 <br>
51 <input type="submit" value="Save" class="btn btn-success">
52 </form>
53 </div>
54
55 <?php
56 // Close Connection
57 mysqli_close($conn);
58 ?>
59
60 </div>
61 </body>
62 </html>
```

“book\_update.php”

```
File Edit Selection View Search
book_update_db.php X
C:\Users\> http://localhost:8080/Courses/Webutvikling/Tutorials/PHP/Development/CRUD Application/Books CRUD Web Application/book_update_db.php

1 <?php
2 require_once 'config.php';
3
4 //Get GET Data
5 $bookid = $_GET["bookid"];
6
7 //Get Post Data
8 $title = $_POST["title"];
9 $author = $_POST["author"];
10 $topic = $_POST["topic"];
11
12 // Insert Data
13 $sql = "UPDATE BOOK SET Title='$title', Author='$author', Topic='$topic' WHERE BookId=$bookid";
14
15 if (mysqli_query($conn, $sql)) {
16     $message = "New book created successfully.<br>";
17 }
18 else {
19     $message = "Error: " . $sql . "<br>" . mysqli_error($conn);
20 }
21
22 // Close Connection
23 mysqli_close($conn);
24 ?>
25
26 <!DOCTYPE html>
27 <html>
28
29 <head>
30 <title>Update Book</title>
31 <meta charset="utf-8">
32 <meta name="viewport" content="width=device-width, initial-scale=1">
33 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
34 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
35 </head>
36
37 <body>
38 <div class="container-fluid pt-5">
39
40 <h1>Update Book</h1>
41
42 <?php echo $message ?><br>
43
44 <a href="books.php" class="btn btn-success">Go back to Books</a>
45
46 </div>
47 </body>
48 </html>
```

Here we save the updated Book information into the Database

“book\_update\_db.php”

<https://www.halvorsen.blog>

CRUD

# Delete Data



Hans-Petter Halvorsen

[Table of Contents](#)

# Updated “books.php”

Books

Here you find a list of available books:

BookId	Title	Author	Topic	Action
1	Web Apps	Elvis Presly	Programming	Update Book : Delete Book
2	IoT and Cloud	John Wayne	IoT	Update Book : Delete Book
3	C#	Rune Hansen	Programming	Update Book : Delete Book
4	AI	Allan Johnsen	Data	Update Book : Delete Book
25	PHP Programming	Hans Hansen	Programming	Update Book : Delete Book
26	DeepSeek	Allan Trumpf	AI	Update Book : Delete Book
36	PHP Programming	Alan Walker	Programming	Update Book : Delete Book

New Book

We add “Delete Book” buttons so we can delete a selected Book:

“books.php”

# Updated “books.php”

We add “Delete Book” buttons so we can delete a selected Book:

“books.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 "<td> <a href='book_update.php?bookid=" . $row["BookId"] . "' class='btn btn-info'>Update Book</a>";
        echo " <a href='book_delete.php?bookid=" . $row["BookId"] . "' class='btn btn-danger'>Delete Book</a> </td>";
        echo "</tr>";
    }
} else {
    echo "0 results";
}
?>
```



# Delete Book

We make a copy of the existing “book\_update\_db.php” and makes the necessary changes:

## Delete Book

Selected book deleted successfully.

Go back to Books

“book\_delete.php”

```
book_delete.php
C:\Users\hansp> OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > CRUD Application > Books CRUD Web Application > book_delete.php
1 <?php
2 require_once 'config.php';
3
4 //Get GET Data
5 $bookid = $_GET["bookid"];
6
7 // Insert Data
8 $sql = "DELETE FROM BOOK WHERE BookId = $bookid";
9
10 if (mysqli_query($conn, $sql)) {
11     $message = "Selected book deleted successfully.<br>";
12 }
13 else {
14     $message = "Error: " . $sql . "<br>" . mysqli_error($conn);
15 }
16
17 // Close Connection
18 mysqli_close($conn);
19 ?>
20
21 <!DOCTYPE html>
22 <html>
23
24 <head>
25 <title>Update Book</title>
26 <meta charset="utf-8">
27 <meta name="viewport" content="width=device-width, initial-scale=1">
28 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
29 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
30 </head>
31
32 <body>
33 <div class="container-fluid pt-5">
34
35 <h1>Delete Book</h1>
36
37 <?php echo $message ?><br>
38
39 <a href="books.php" class="btn btn-success">Go back to Books</a>
40
41 </div>
42 </body>
43 </html>
```

# Improvement

“books.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 "<td> <a href='book_update.php?bookid=" . $row["BookId"] . "' class='btn btn-info'>Update Book</a>";
        //echo " <a href='book_delete.php?bookid=" . $row["BookId"] . "' class='btn btn-danger'>Delete Book</a> </td>";
        echo " <a href='javascript:deleteBook(" . $row["BookId"] . ")' class='btn btn-danger'>Delete Book</a> </td>";
        echo "</tr>";
    }
} else {
    echo "0 results";
}
?>
```



Before we Delete a Book, we want to ask the user if the user really want to Delete the Book. Let's add some **JavaScript** code.

JavaScript Function

```
18 function deleteBook(bookid)
19 {
20     let message = "Do you really want to delete book #" + bookid + "?";
21
22     if (confirm(message) == true)
23     {
24         window.location.href = "book_delete.php?bookid=" + bookid;
25     }
26 }
27
28 </script>
```

<https://www.halvorsen.blog>

# Adding Security



Hans-Petter Halvorsen

[Table of Contents](#)

<https://www.halvorsen.blog>

# Summary and Further Work



Hans-Petter Halvorsen

[Table of Contents](#)

# Summary

We have created the following PHP Web Application with basic CRUD functionality:

## Books

Here you find a list of available books:

Bookid	Title	Author	Topic	Action
1	Web Apps	Elvis Presly	Programming	<a href="#">Update Book</a> <a href="#">Delete Book</a>
2	IoT and Cloud	John Wayne	IoT	<a href="#">Update Book</a> <a href="#">Delete Book</a>
3	C#	Rune Hansen	Programming	<a href="#">Update Book</a> <a href="#">Delete Book</a>
4	AI	Allan Johnsen	Data	<a href="#">Update Book</a> <a href="#">Delete Book</a>

[New Book](#)

Do you really want to delete book #'1'?

[OK](#)

[Cancel](#)

## New Book

Please enter book information:

Title:

Author:

Topic:

[Save](#)

Now you should be able to create your own PHP CRUD Web Application

## Update Book

Please enter book information:

Title:

Author:

Topic:

[Save](#)

I encourage you to improve the simplified example and add more functionality, more robustness, etc.

# Suggested Improvements

- Improve **Database** structure, create more tables and relations, create and use views, stored procedures, etc
- Improve **Security**, i.e., use more of the built-in features in PHP
- Add **Login** functionality. Meaning the user needs to login when starting the application. Then for every page we need to check if the user has logged in or not. For this purpose, we can use a Session variable.
- Improve Code Quality and structure, i.e., make and use Classes, etc.
- Improve **Robustness** and add **Error Handling**, etc.
- Make the app more **User-friendly** and intuitive
- Improve **GUI** and Layout
- Add more Features
- ...

# Adding Security

PHP has many built-in features for making your apps more secure

- Use the function “`htmlspecialchars()`”
  - This function is used to convert special characters to HTML entities. This is particularly useful for preventing cross-site scripting attacks.
- You should also use the “`trim()`” and “`stripslashes()`” functions

# Adding Security

```
$bookid = $_GET["bookid"];
```



Create e.g., a function like this:

```
function checkInput($data) {  
    $data = trim($data);  
    $data = stripslashes($data);  
    $data = htmlspecialchars($data);  
    return $data;  
}
```



```
$bookid = checkInput($_GET["bookid"]);
```

Since this is a PHP function that you will use in many different PHP files you can put this function in a separate file called, e.g., “functions.php” that you include in your PHP files.



# Use “required”

```
<label for="title" class="form-label">Title:</label>  
<input type="text" id="title" name="title" class="form-control">
```



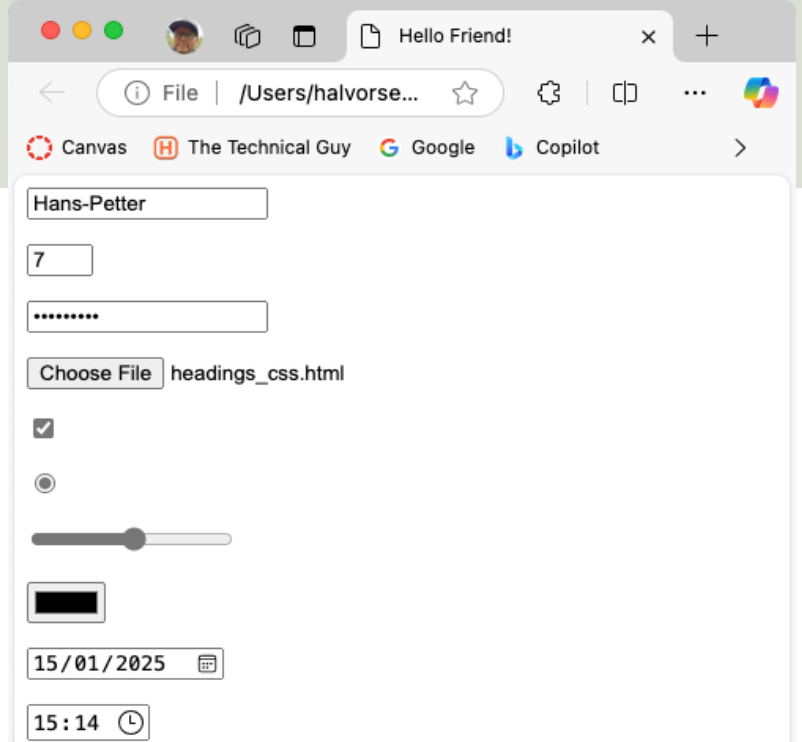
```
<label for="title" class="form-label">Title:</label>  
<input type="text" id="title" name="title" class="form-control" autofocus required>
```

For fields that must be filled out by the user, you should use “required”

Here is also “autofocus” used to improve the user experience.

# HTML Input Types

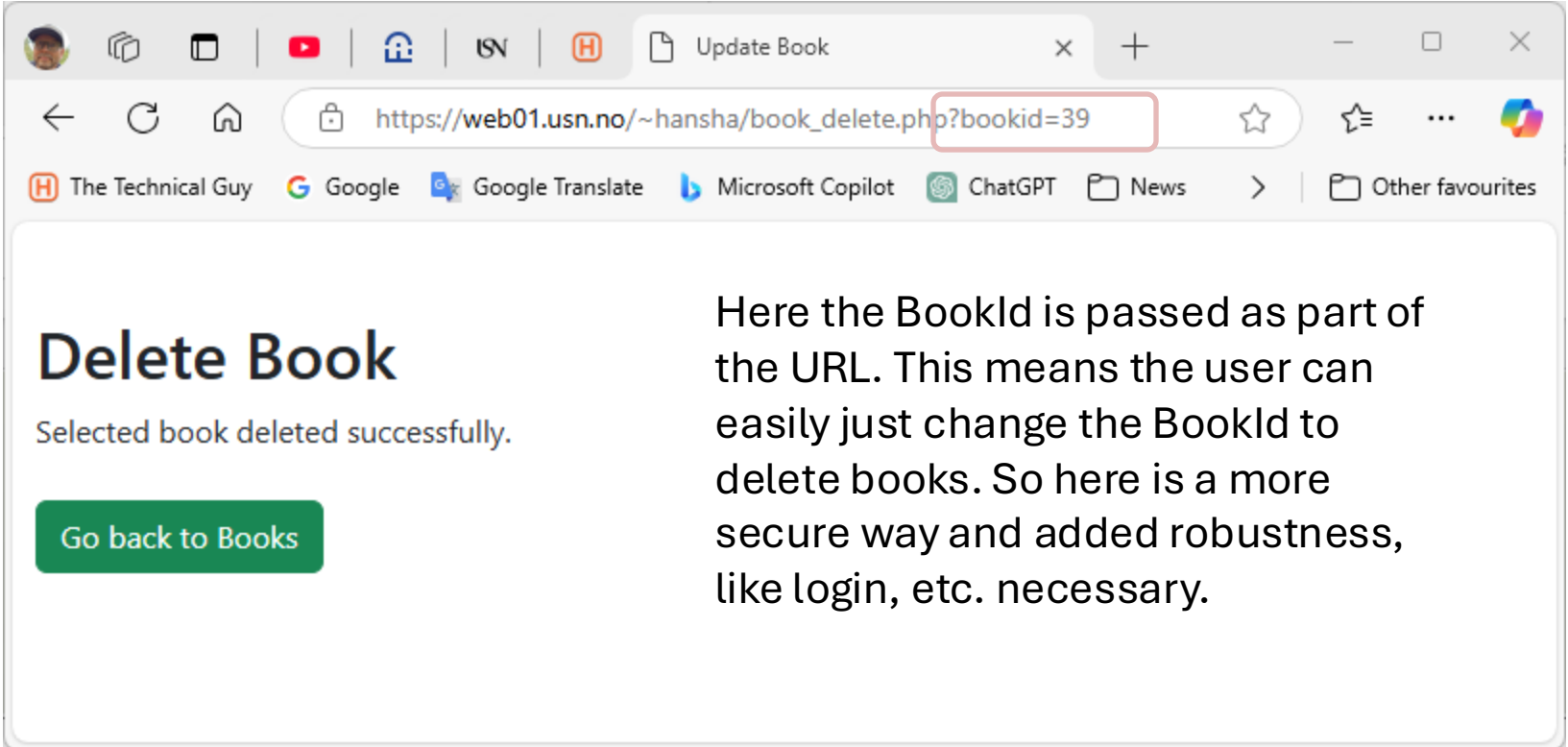
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Hello Friend!</title>
5   </head>
6   <body>
7
8     <input type="text" value="Hans-Petter"><p></p>
9     <input type="number" min="1" max="10"><p></p>
10    <input type="password"><p></p>
11    <input type="file"><p></p>
12
13    <input type="checkbox"><p></p>
14    <input type="radio"><p></p>
15    <input type="range"><p></p>
16    <input type="color"><p></p>
17
18    <input type="date"><p></p>
19    <input type="time"><p></p>
20    <input type="datetime-local"><p></p>
21    <input type="month"><p></p>
22
23    <input type="button" value="Button"><p></p>
24    <input type="submit">
25
26  </body>
27 </html>
```



Make sure to use proper input types and not “text” for everything. If you expect a number, use type=“number”. If you expect an email address, use type=“email”. Etc.

Reference: [https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)

# Query String



The screenshot shows a web browser window with the following elements:

- Browser Tab:** "Update Book"
- Address Bar:** `https://web01.usn.no/~hansha/book_delete.php?bookid=39`. The query string `?bookid=39` is highlighted with a red box.
- Bookmarks:** The Technical Guy, Google, Google Translate, Microsoft Copilot, ChatGPT, News, and Other favourites.
- Page Content:**
  - ## Delete Book
  - Selected book deleted successfully.
  - [Go back to Books](#) (button)
- Text on the right:** "Here the BookId is passed as part of the URL. This means the user can easily just change the BookId to delete books. So here is a more secure way and added robustness, like login, etc. necessary."

# Create User and Login

This is done  
in another  
Tutorial!

It is recommended that you create login functionality. Here is a simple example:

1. Create a new Database Table with Username and Password information.
2. Create a “NewUser“ page (e.g., “new\_user.php”) that insert UserName and Password information into the Database.
  - Here you can use the built-in “**password\_hash**” PHP function.
3. Create a “Login” page (e.g., “login.php”) where the user needs to enter UserName and Password.
  - Check the entered UserName and Password with the information stored in the Database. Here you can use the built-in “**password\_verify**” PHP function .
  - If Password is correct, create a Session variable that says the user is logged in.
4. In all other PHP files, perform a check in the beginning whether the user is logged in or not (check if the session variable is true)

# 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](http://www.usn.no)

E-mail: [hans.p.halvorsen@usn.no](mailto:hans.p.halvorsen@usn.no)

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

