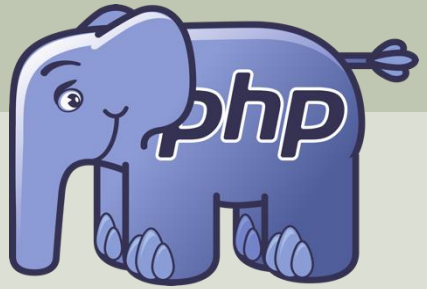
https://www.halvorsen.blog



PHP and MySQL CRUD Web Application

Step by Step Instructions

Hans-Petter Halvorsen

Contents

Introduction

Hans-Petter Halvorsen

Table of Contents

Introduction

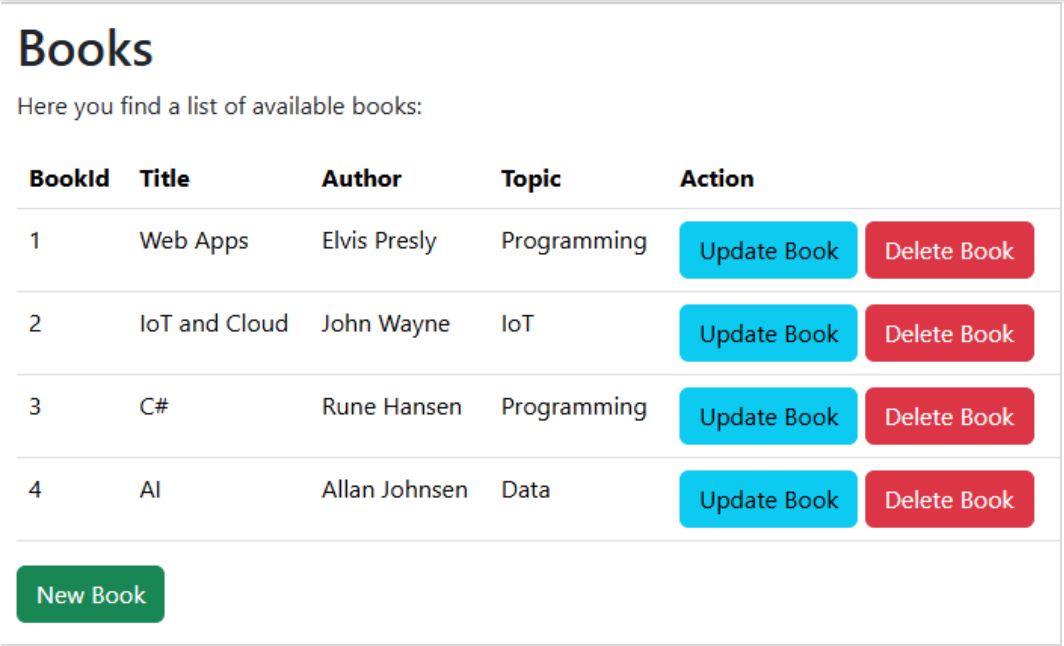
- We will create a CRUD Web Application using PHP and MySQL.
- CRUD means Create, Read, Update and Delete.
- This means our Web Application should retrieve data, insert data, update data and delete data from a MySQL database.
- We will go through and create the web application step by step by adding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

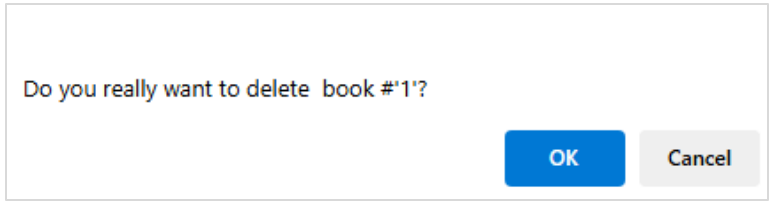
Introduction

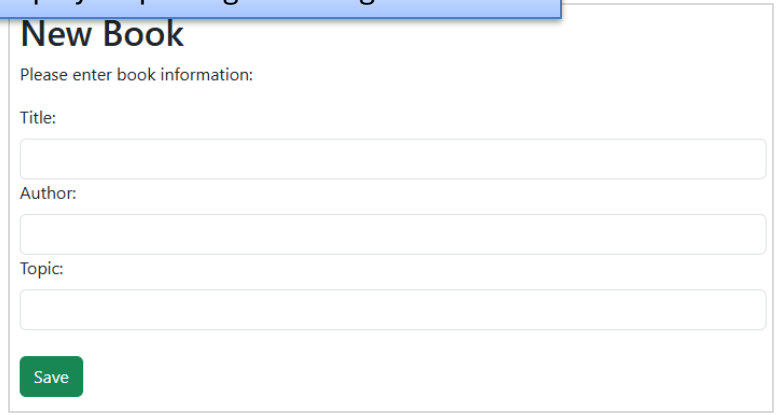
- MySQL is a popular relational database system.
- It is free and open source.
- MySQL uses SQL (Structured Query Language)
- The combination of PHP and MySQL is very popular.
- You can create powerful web applications where you can show, save, update and delete data in a MySQL database from the PHP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

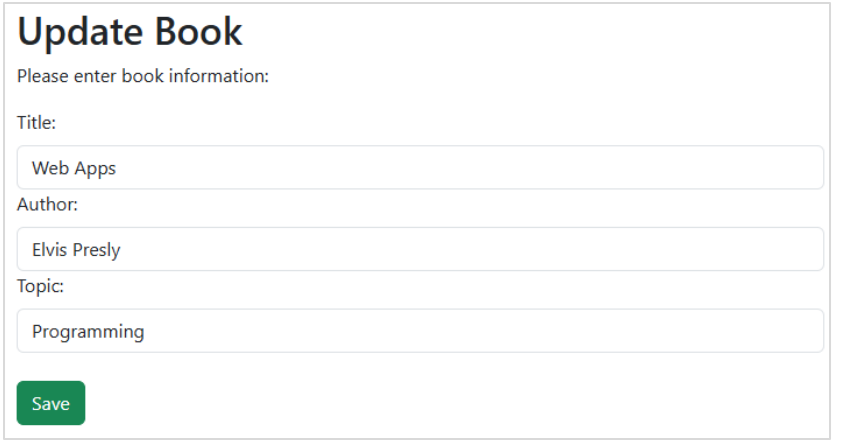
Books Web Application

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









Tools

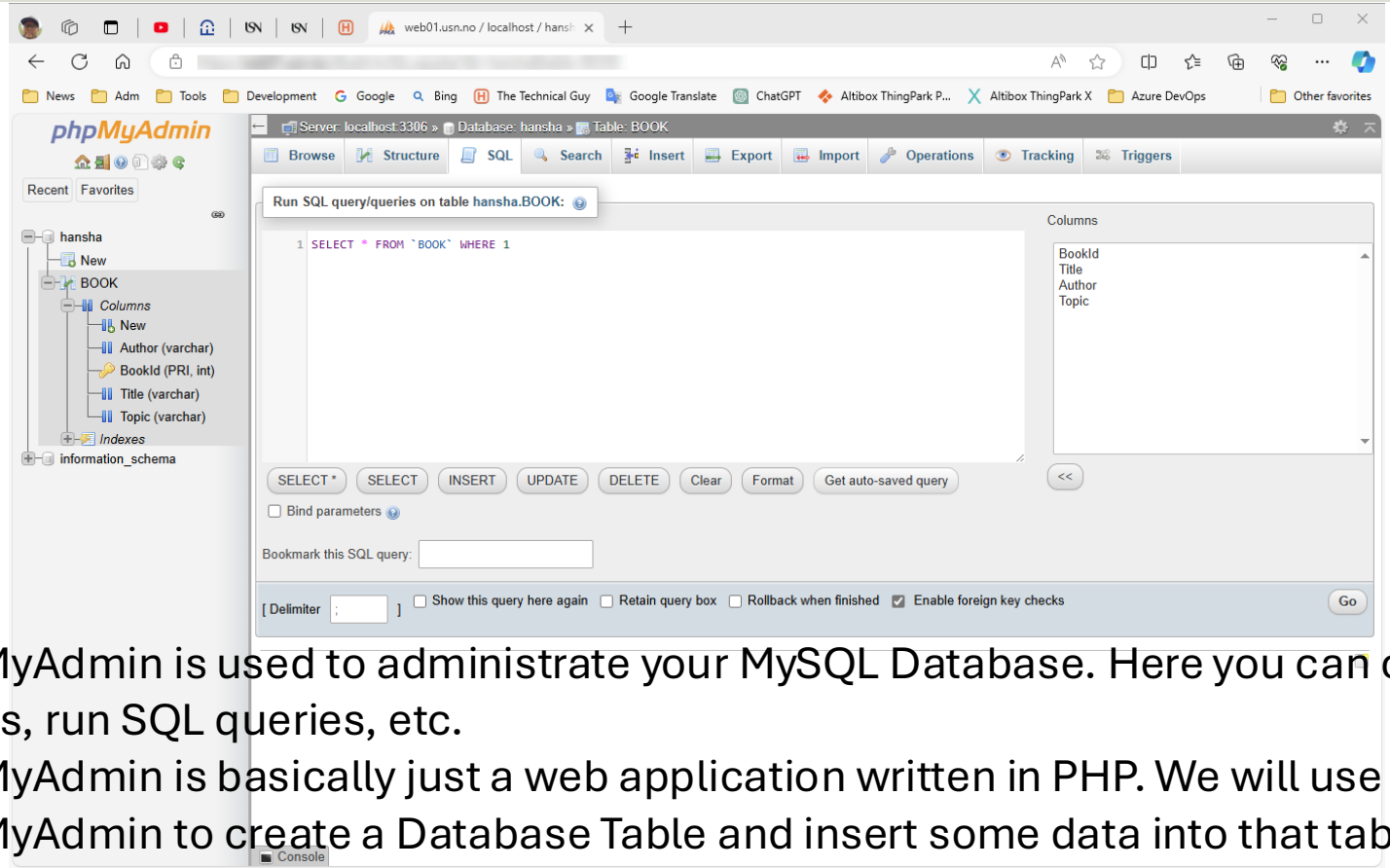
- **PHP** - a server scripting language for making dynamic web pages, typically communicating with a Database.
- We will host our PHP files on an existing **Web Server** that supports PHP and MySQL. You can also create your own.
- We will use **Visual Studio Code** (you can use another IDE if you prefer).
- 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).

Database

Hans-Petter Halvorsen

Table of Contents

phpMyAdmin

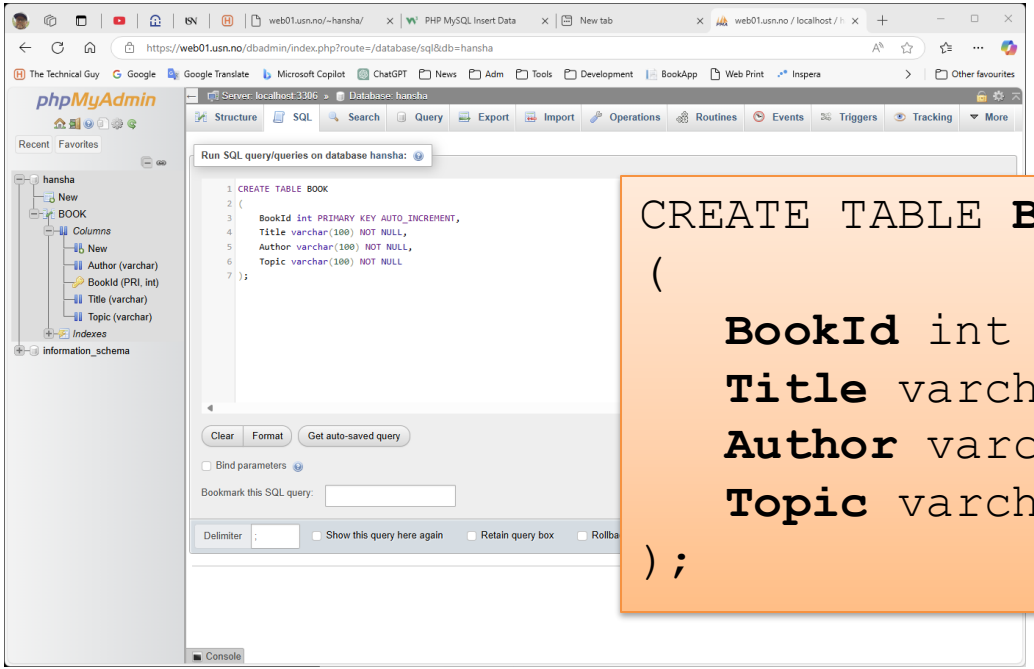


phpMyAdmin is used to administrate your MySQL Database. Here you can create tables, run SQL queries, etc.

phpMyAdmin is basically just a web application written in PHP. 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');
```

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
<?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

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

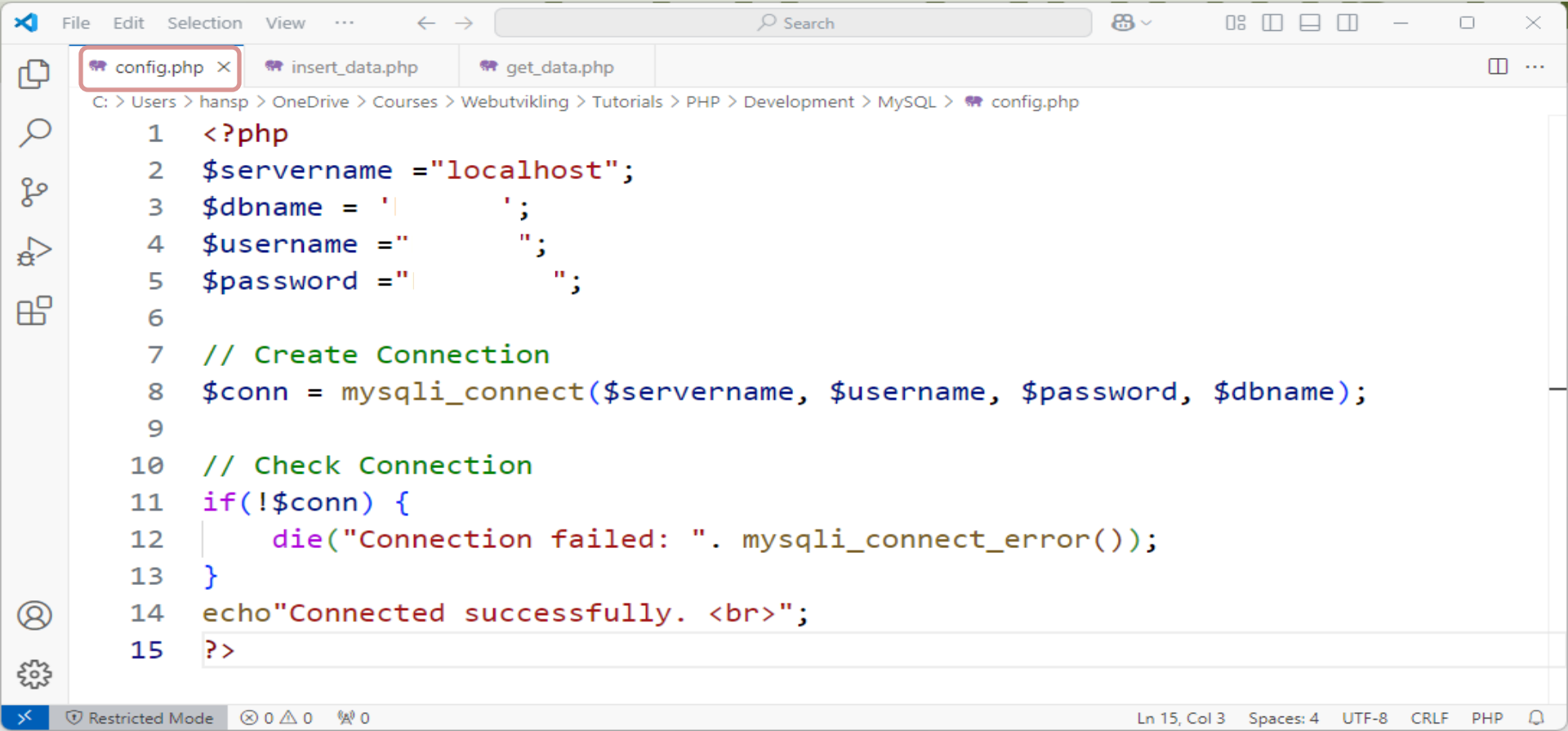
PHP Config File

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

config.php

```php
<?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.";
?>
```

PHP Config File



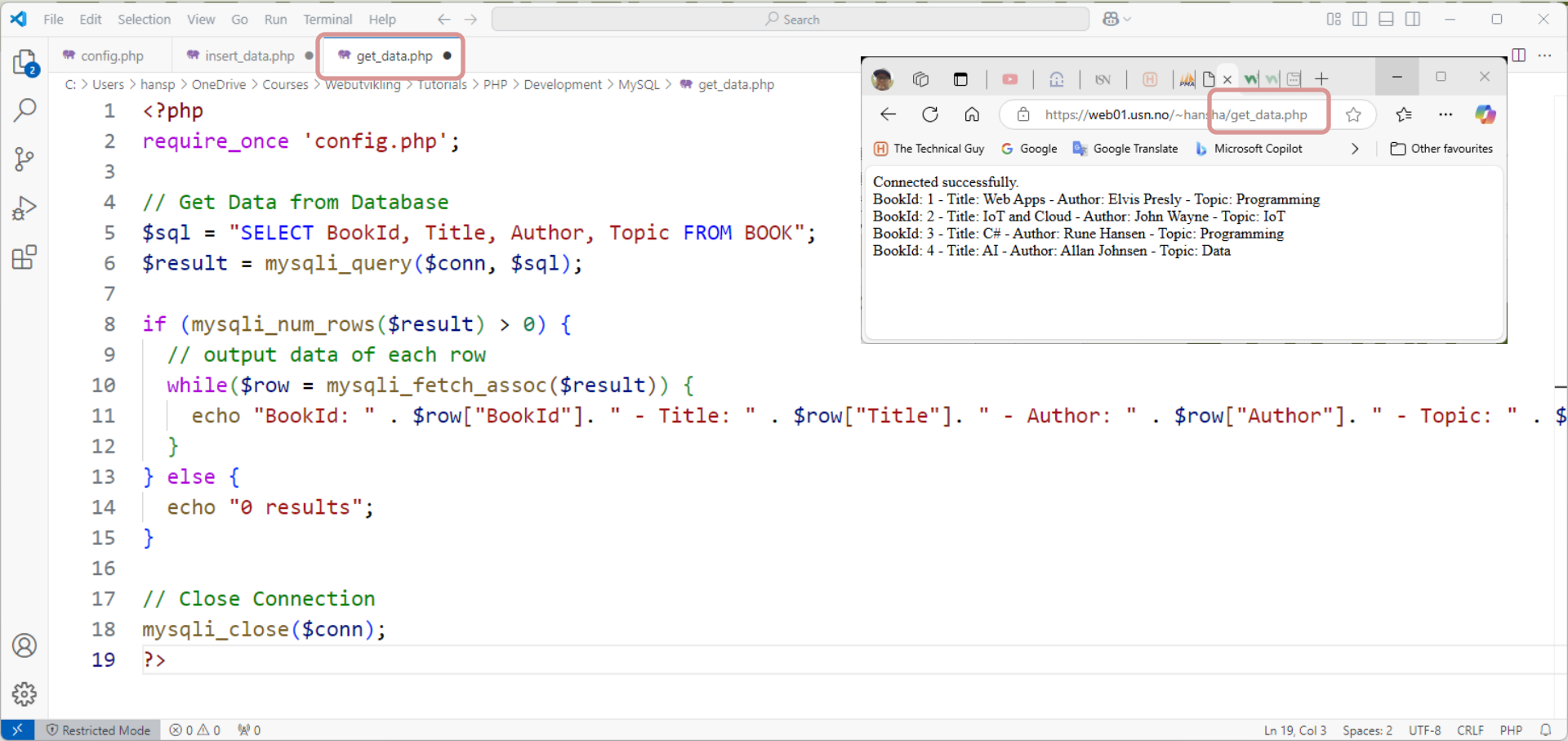
CRUD

Read Data

Hans-Petter Halvorsen

Table of Contents

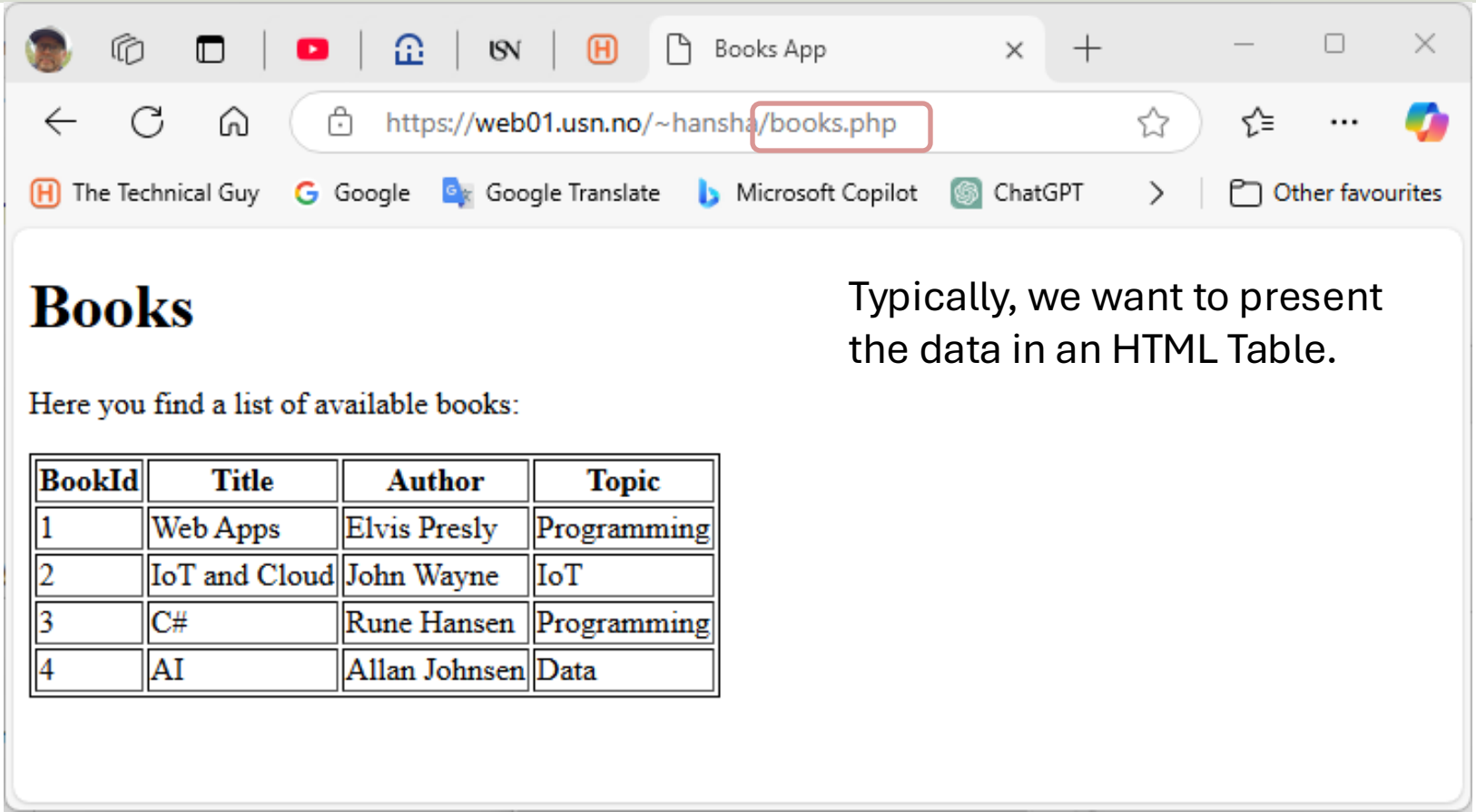
Show Data from the Database



Code

```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);
?>
```

Present Data in an HTML Table



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

```php
<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
<?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>
```

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">
```

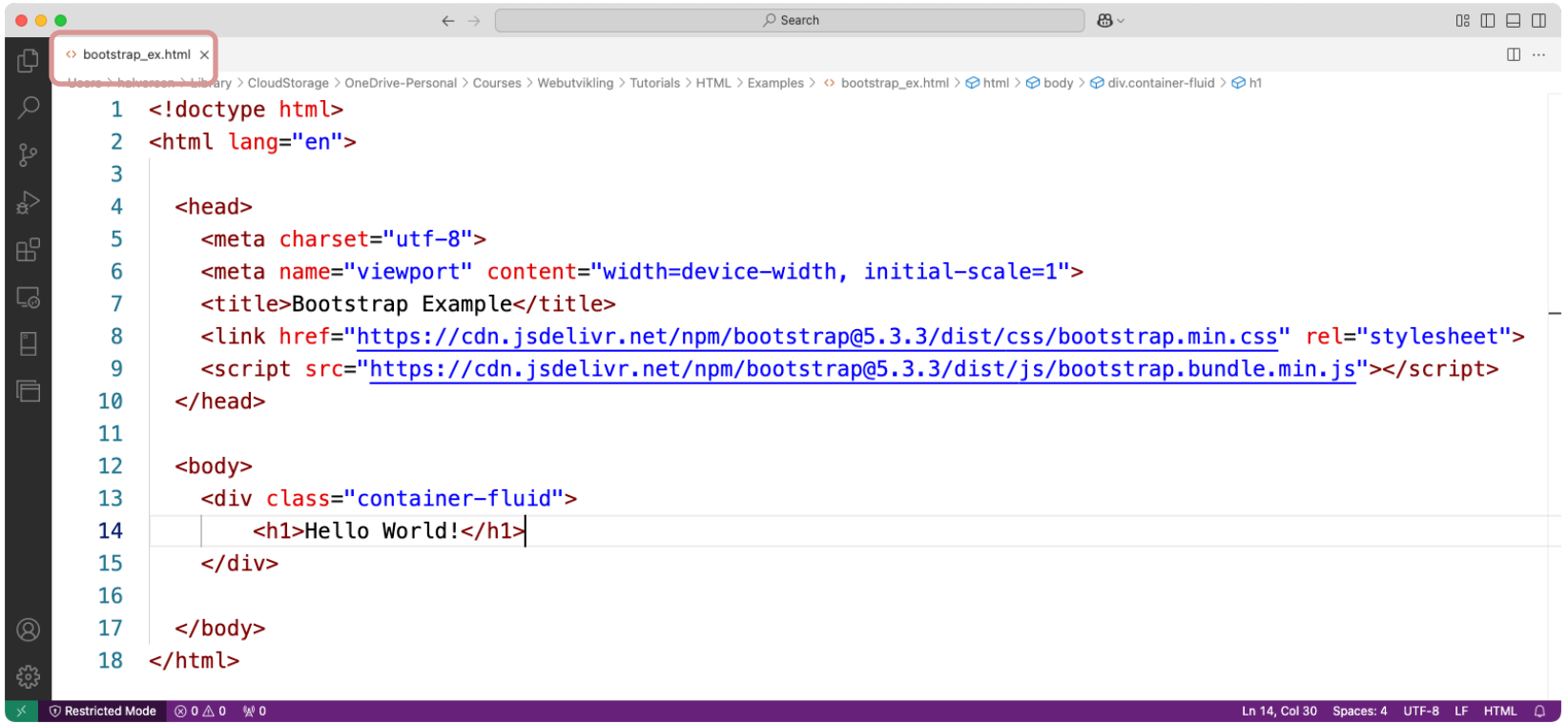
2. Bootstrap JavaScript Library :

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

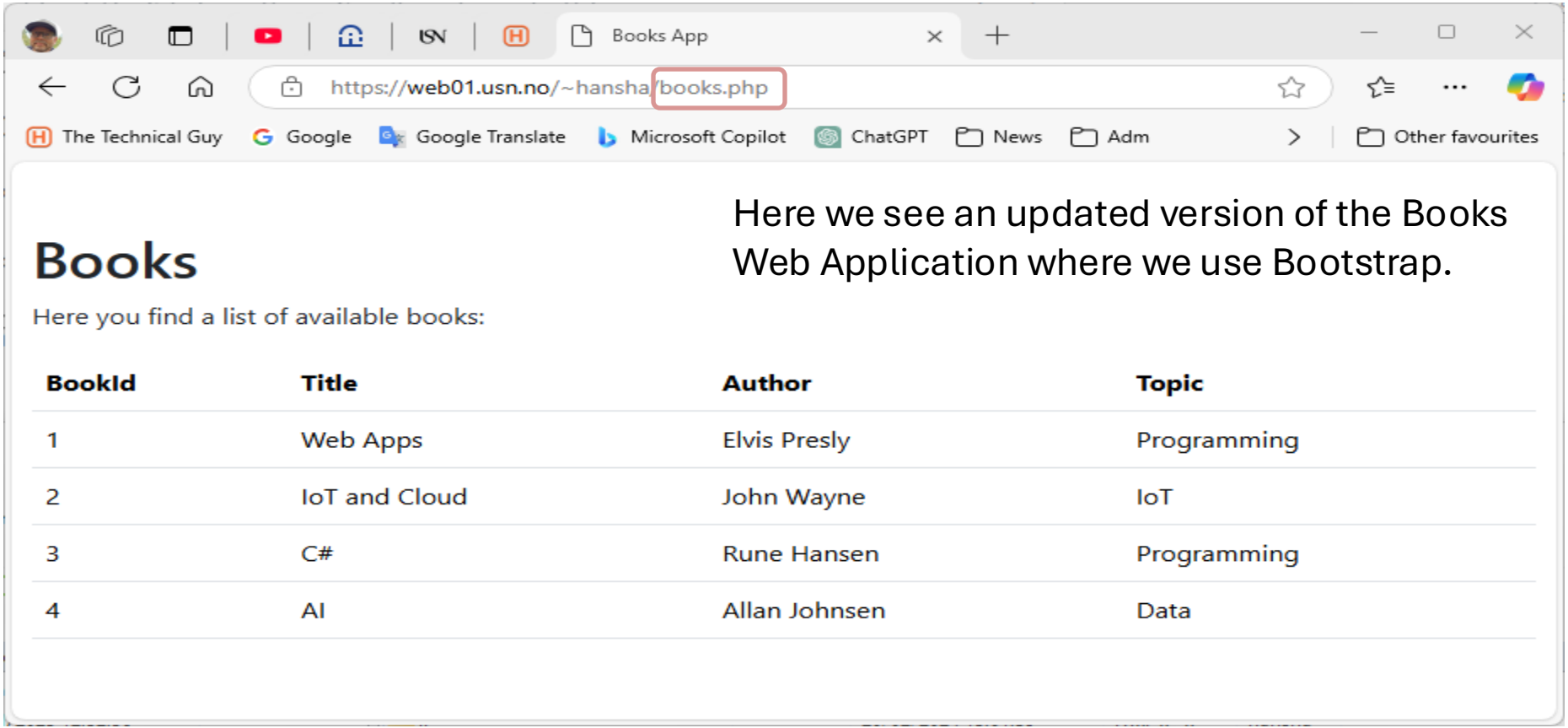
3. This line of code ensures your code works fine on all devices like smartphones, etc:

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

Bootstrap - Hello World Example



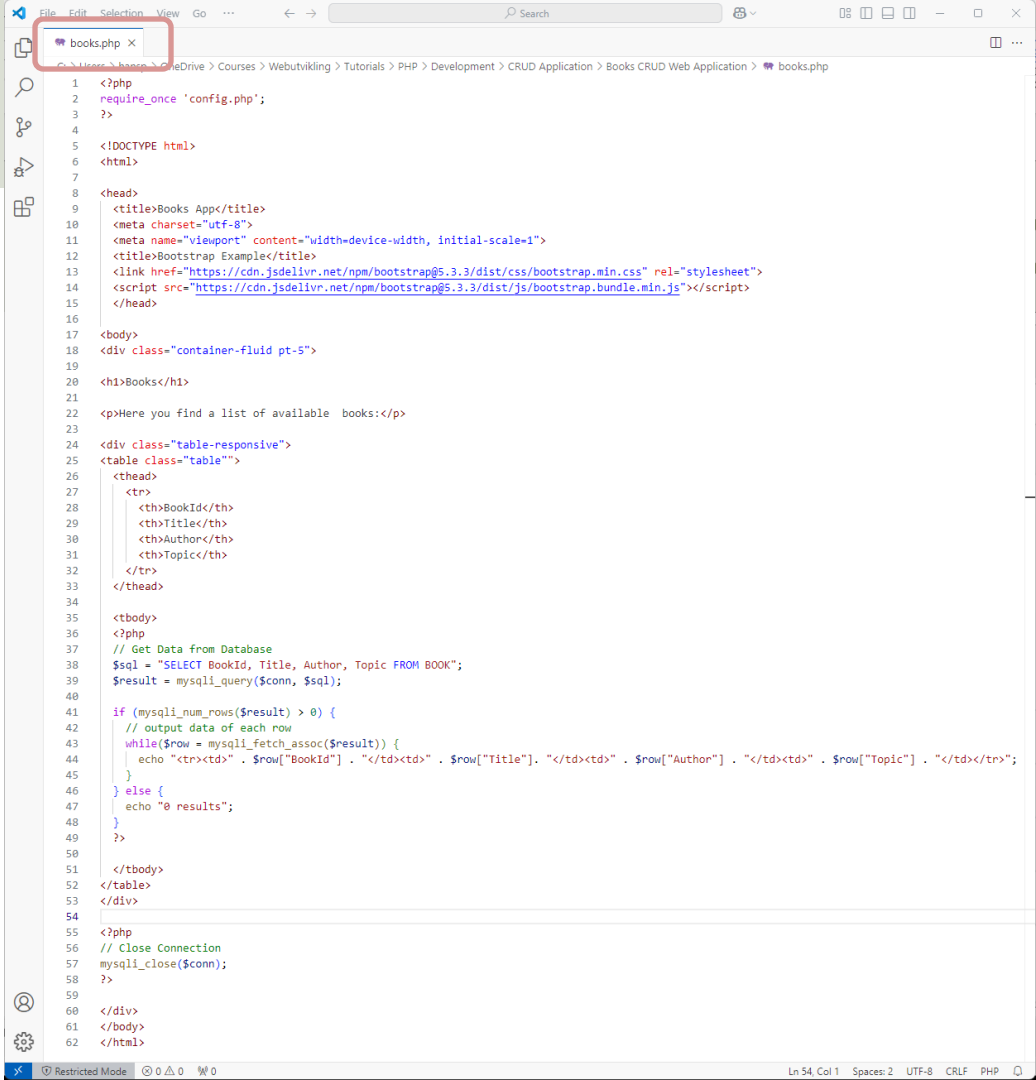
Updated Books Application



Here we see an updated version of the Books Web Application where we use Bootstrap.

Updated Code

“books.php”



CRUD

Create Data

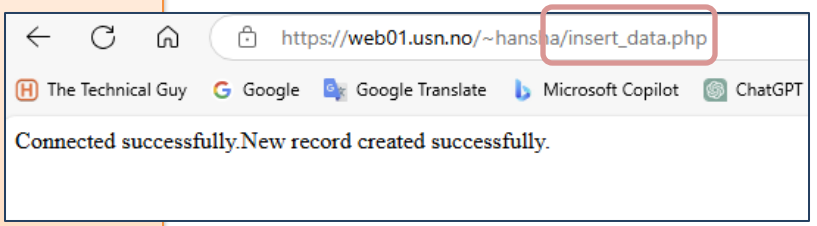
Table of Contents

Hans-Petter Halvorsen

Save Data to the Database

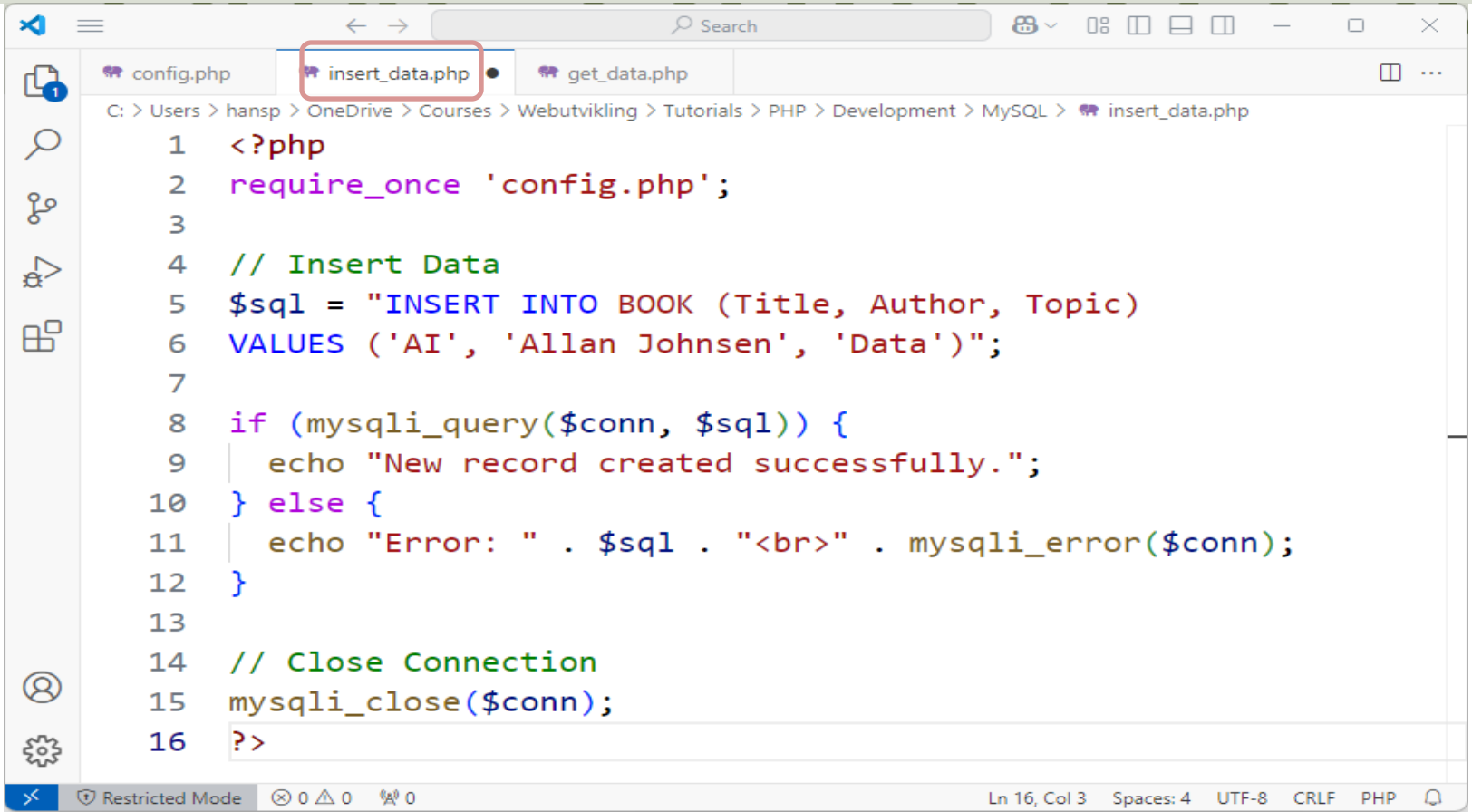
```php
<?php
require_once 'config.php';
// Insert Data
$sql = "INSERT INTO BOOK (Title, Author, Topic)
VALUES ('AI', 'Allan Johnsen', 'Data')";

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

"insert_data.php"

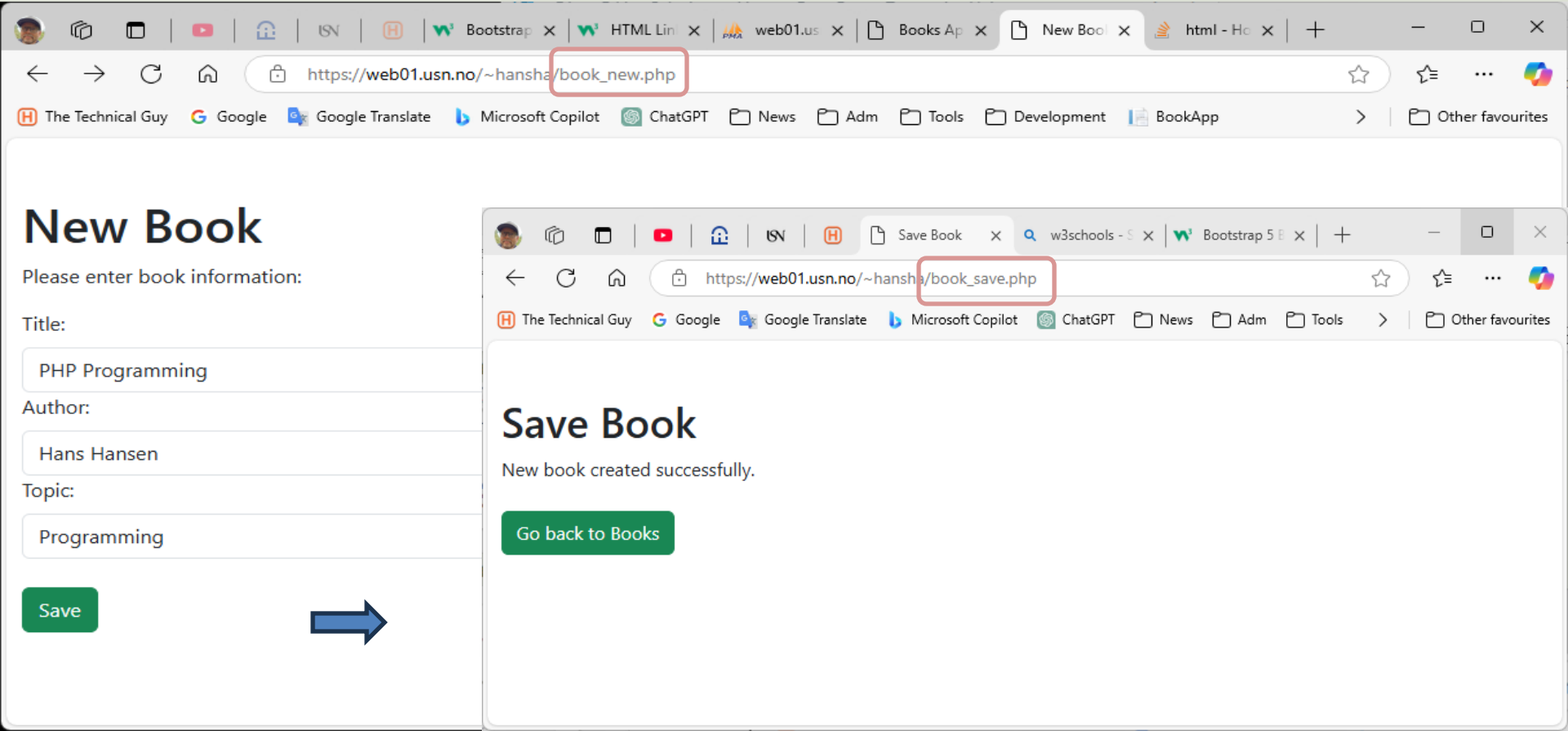


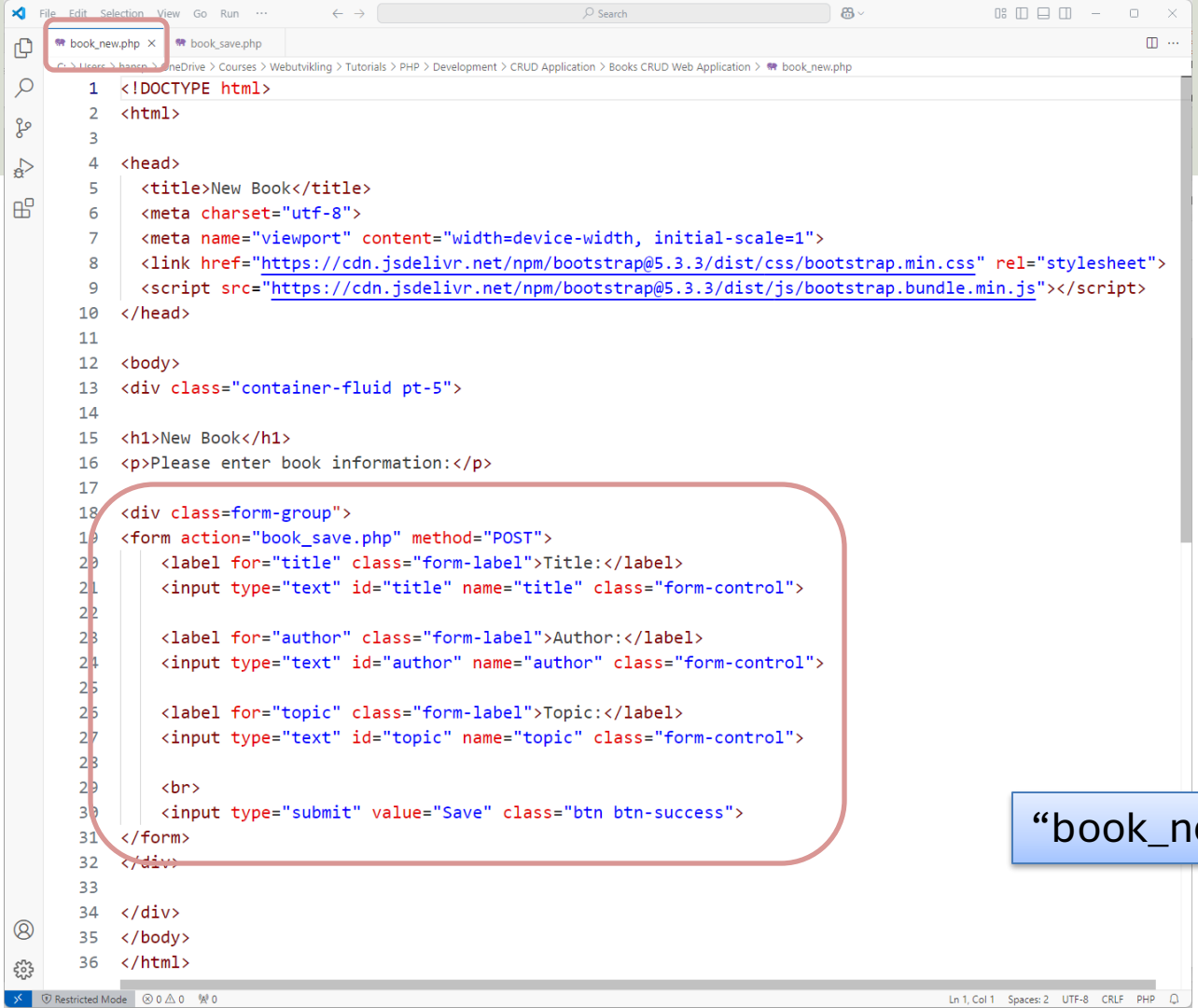
Then go to **phpMyAdmin** and check if the data has been stored in the database.

Save Data to the Database

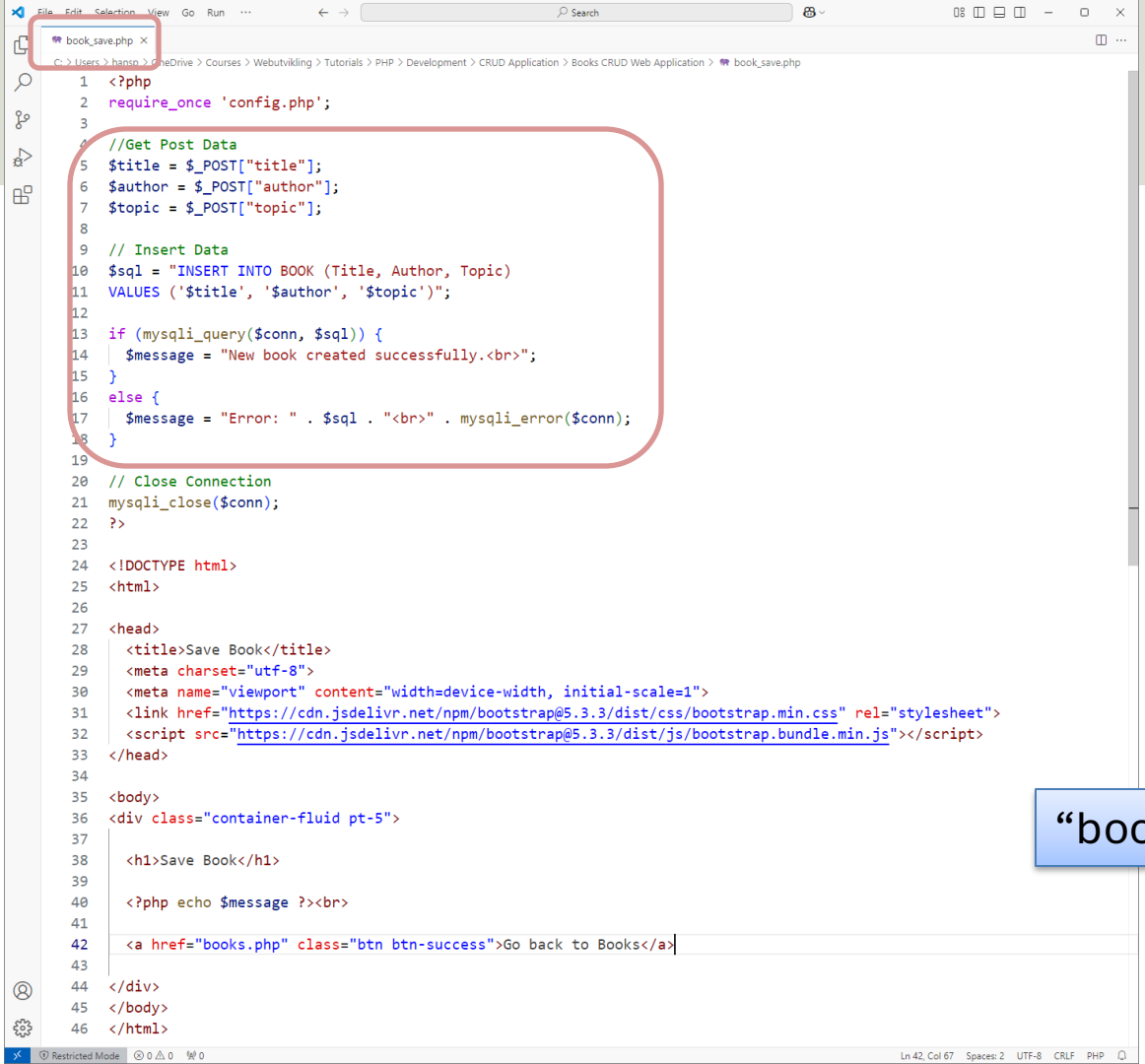


Create an HTML Form




book_new.php
book_save.php
Users > hansp > OneDrive > 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
14
15 <h1>New Book</h1>
16 Please enter book information:
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
30 <input type="submit" value="Save" class="btn btn-success">
31 </form>
32
33
34
35 </body>
36 </html>
Restricted Mode
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
PHP

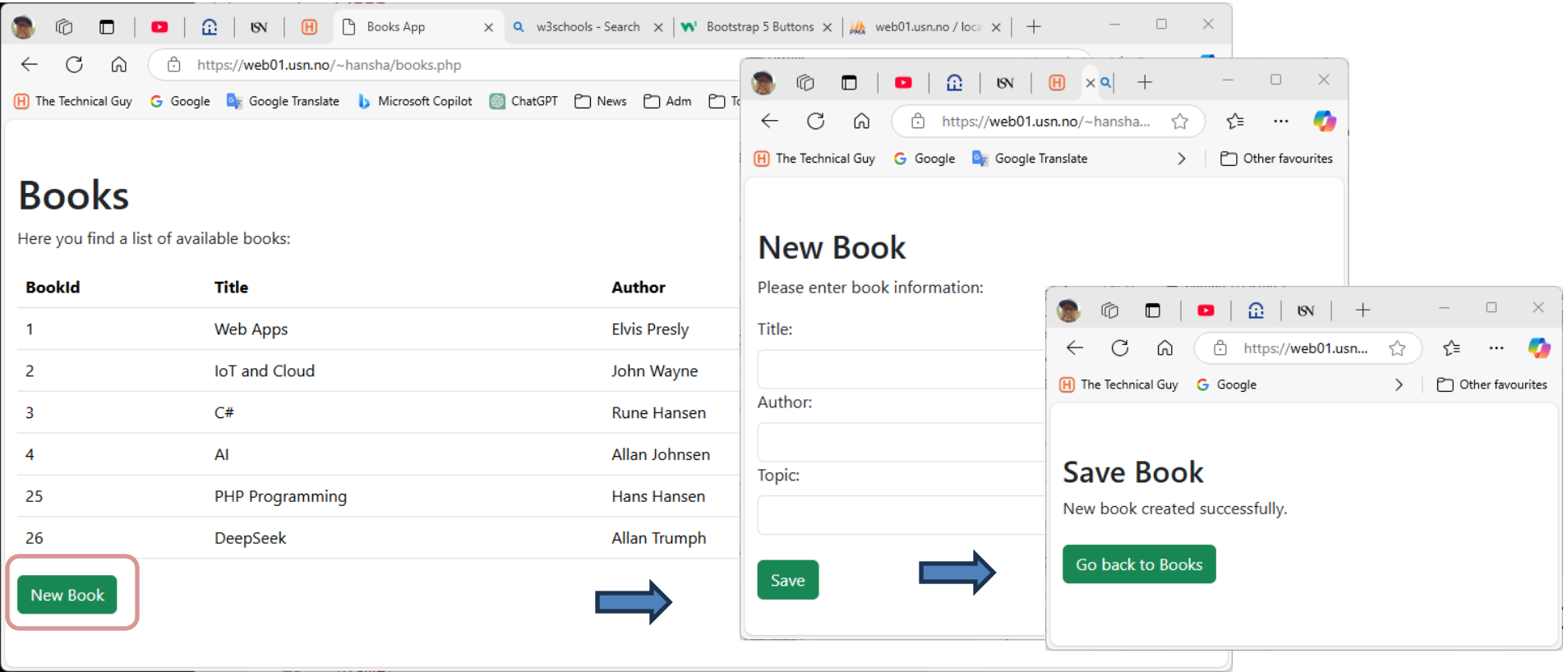
“book_new.php”



“book_save.php”

Updated “books.php”

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



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
```

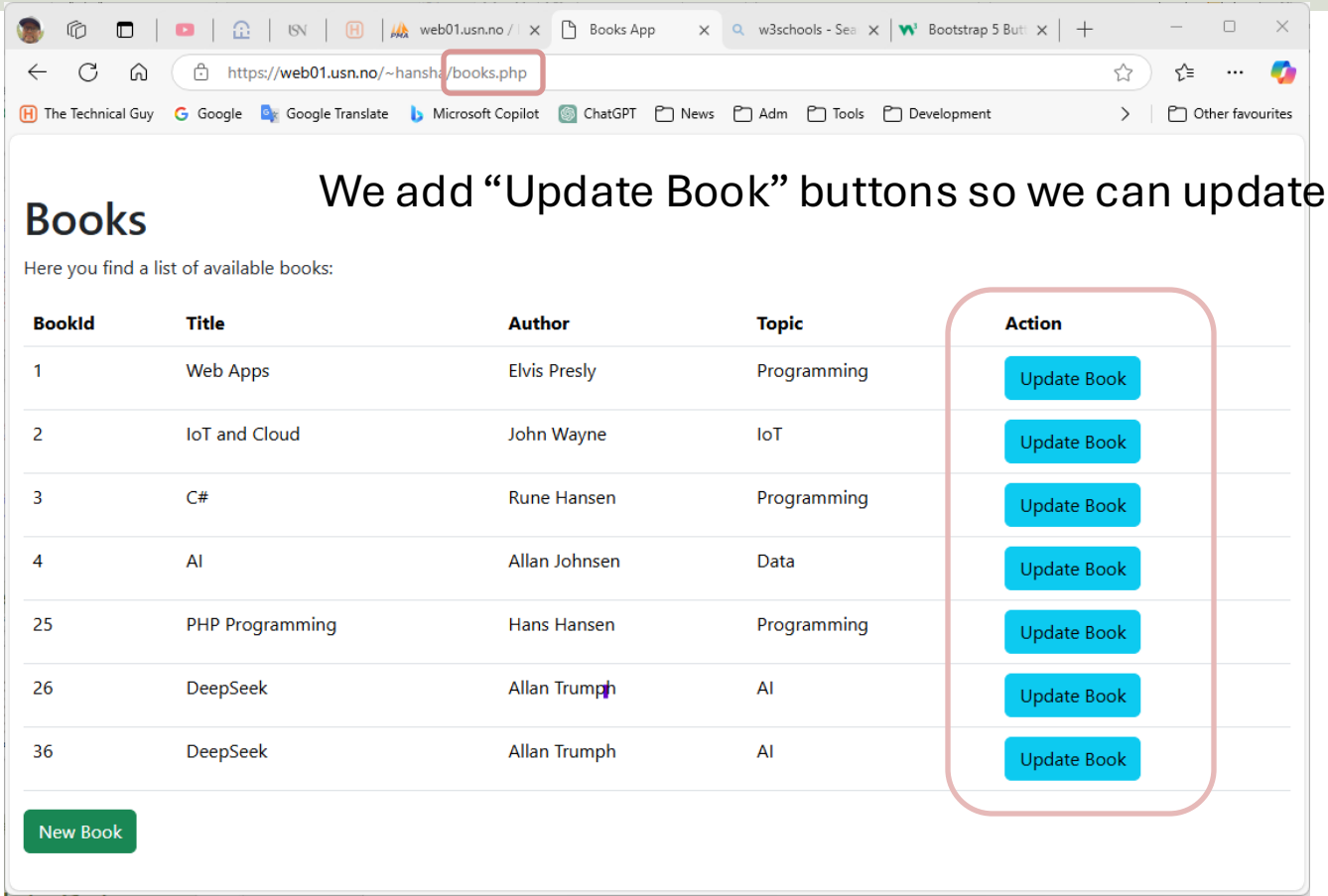
CRUD

Update Data

Hans-Petter Halvorsen

Table of Contents

Updated “books.php”



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

```
<table class="table"">
  <thead>
    <tr>
      <th>BookId</th>
      <th>Title</th>
      <th>Author</th>
      <th>Topic</th>
      <th>Action</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 "<td> <a href='book_update.php?bookid=" . $row["BookId"] . "' class='btn btn-info'>Update Book</a> </td>";
      echo "</tr>";
    }
  } else {
    echo "0 results";
  }
  ?>

  </tbody>
</table>
```

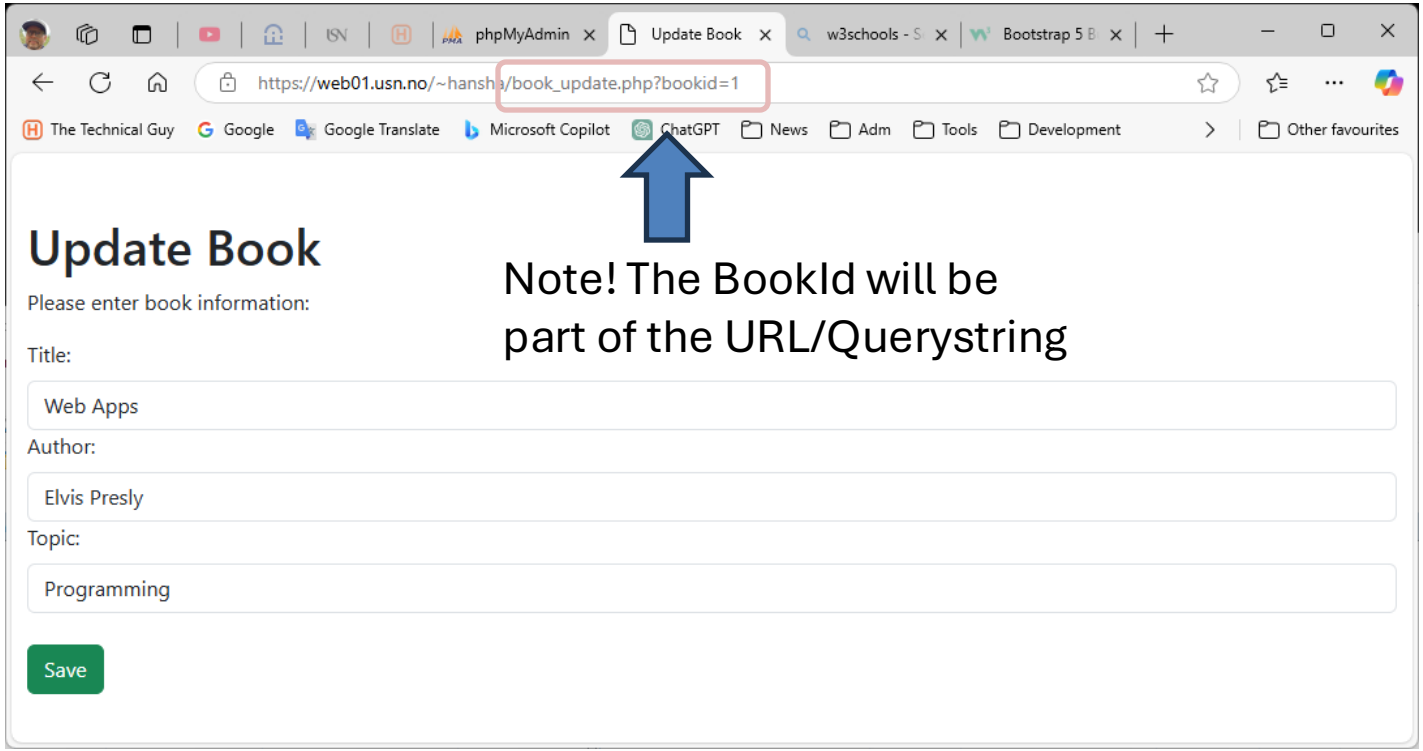
We add a “Update Book” buttons so we update a selected Book:

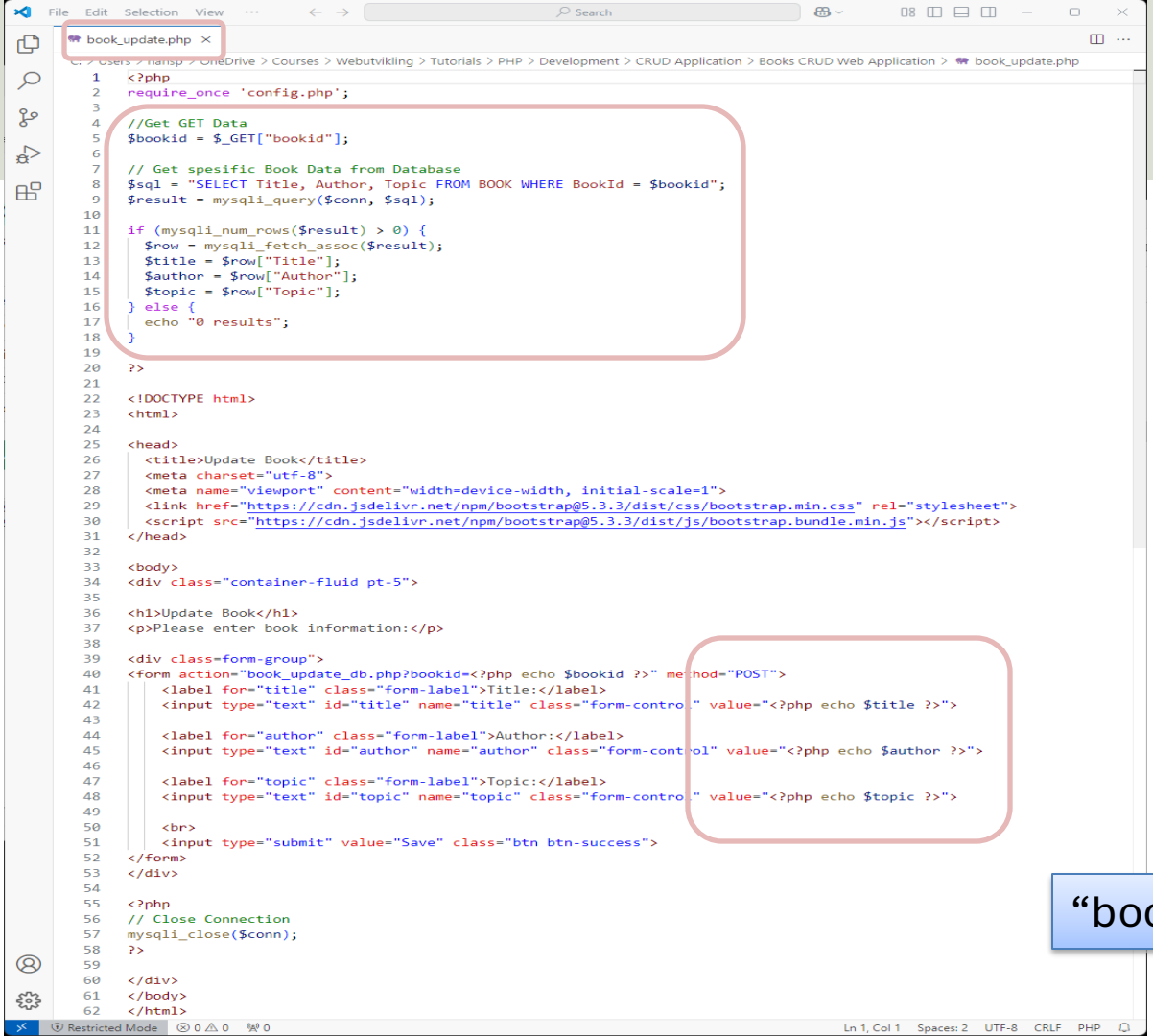
We add a Hyperlink for each row

Each Book has a unique BookId

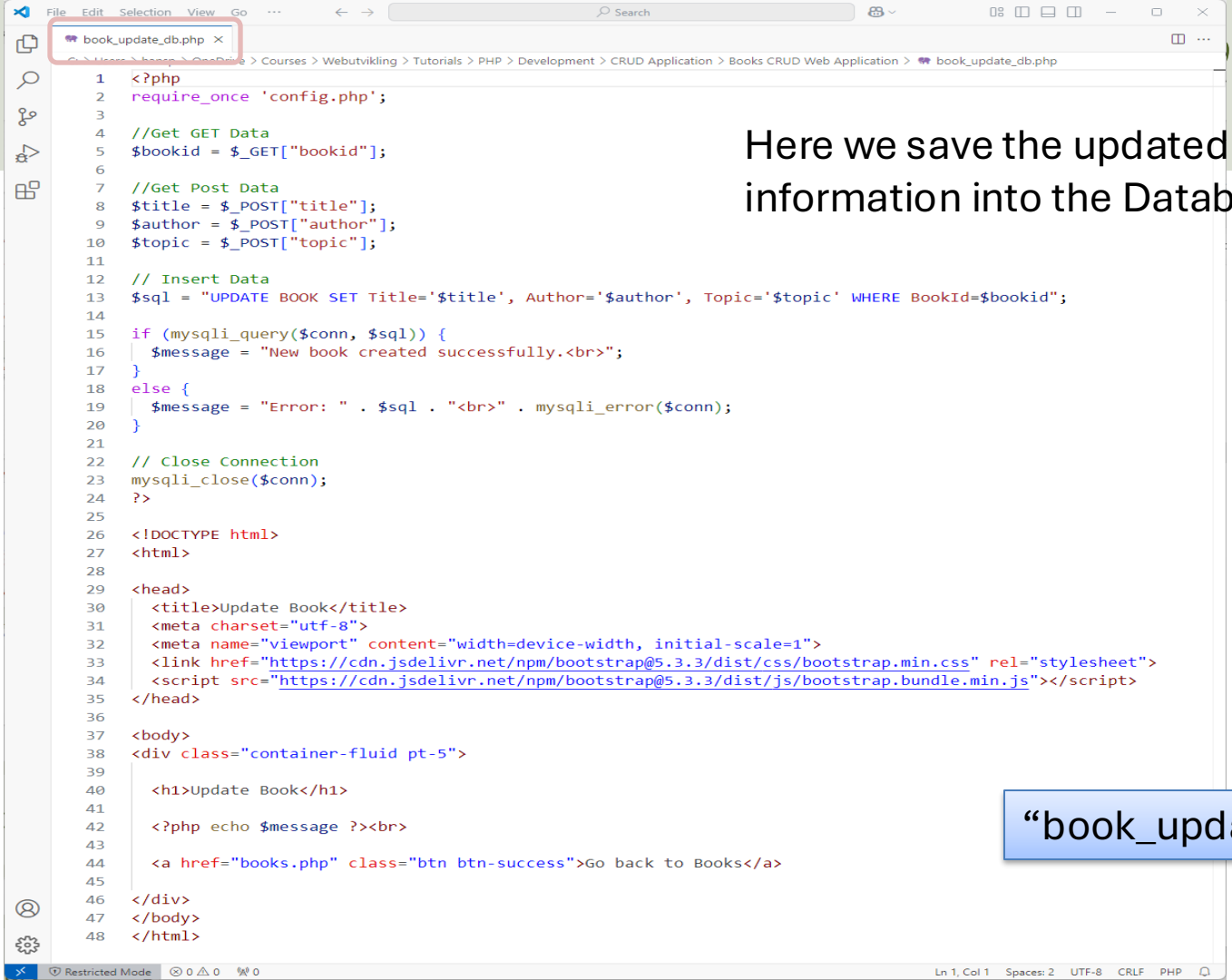
We create “book_update.php”

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





"book_update.php"



Here we save the updated Book information into the Database

“book_update_db.php”

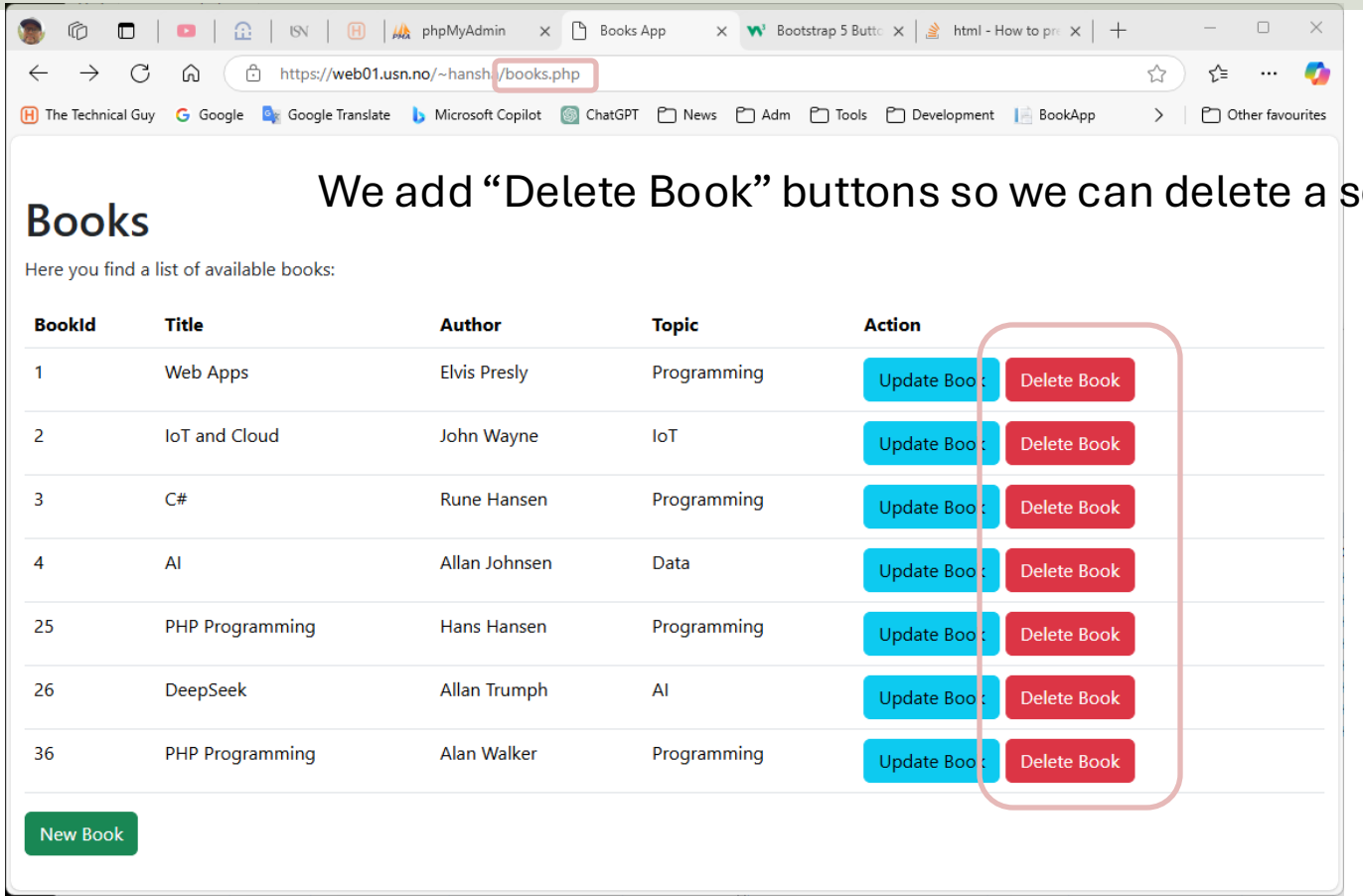
CRUD

Delete Data

Hans-Petter Halvorsen

Table of Contents

Updated “books.php”



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

“books.php”

Updated “books.php”

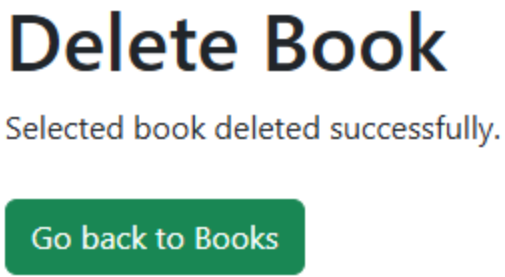
We add “Delete Book” buttons so we can delete a selected Book:

“books.php”

```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:



“book_delete.php”

book_delete.php

C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > CRUD Application > Books CRUD Web Application > book_delete.php

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

//Get GET Data
$bookid = $_GET["bookid"];

// Insert Data
$sql = "DELETE FROM BOOK WHERE BookId = $bookid";

if (mysqli_query($conn, $sql)) {
  $message = "Selected book deleted successfully.<br>";
}
else {
  $message = "Error: " . $sql . "<br>" . mysqli_error($conn);
}

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

<!DOCTYPE html>
<html>

<head>
  <title>Update Book</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
</head>

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

  <h1>Delete Book</h1>

  <?php echo $message ?><br>

  <a href="books.php" class="btn btn-success">Go back to Books</a>

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

Improvement

"books.php"

```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

```javascript
function deleteBook(bookid)
{
  let message = "Do you really want to delete  book #'" + bookid + "'?";

  if (confirm(message) == true)
  {
    window.location.href = "book_delete.php?bookid=" + bookid;
  }
}

</script>
```

Adding Security

Hans-Petter Halvorsen

Table of Contents

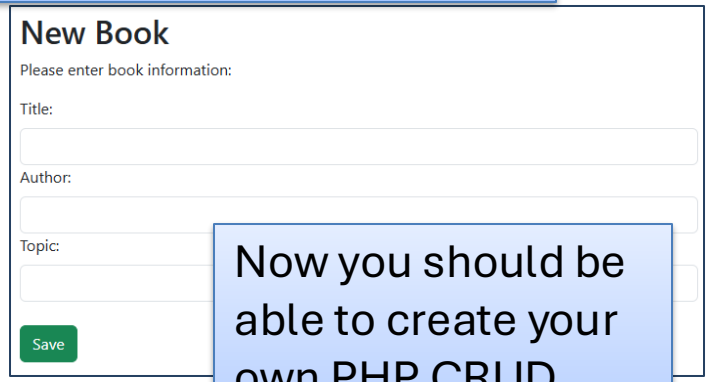
Summary and Further Work

Table of Contents

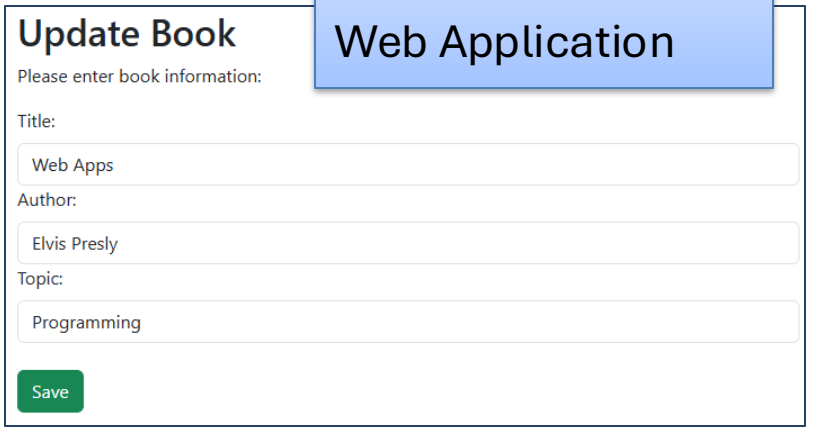
Hans-Petter Halvorsen

Summary

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



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



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

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

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

Reference: https://www.w3schools.com/php/php_form_validation.asp

Use “required”

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

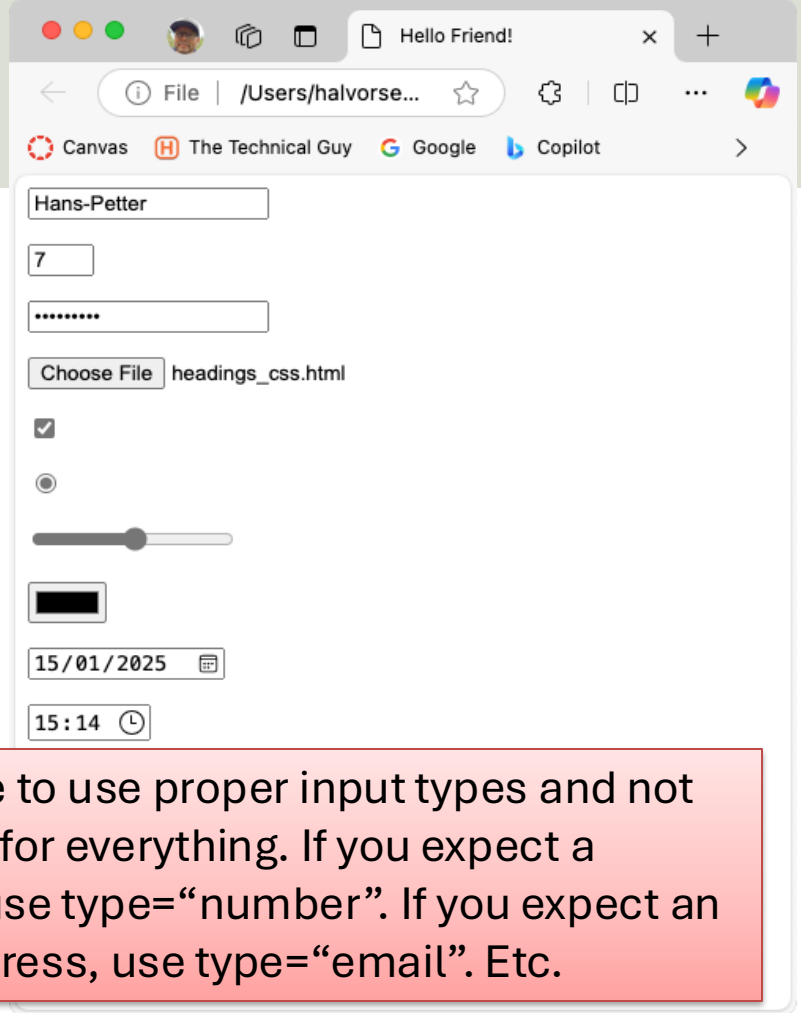
```
<label for="title" class="form-label">Title:</label>
<input type="text" id="title" name="title" class="form-control" autofocus required>
```

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

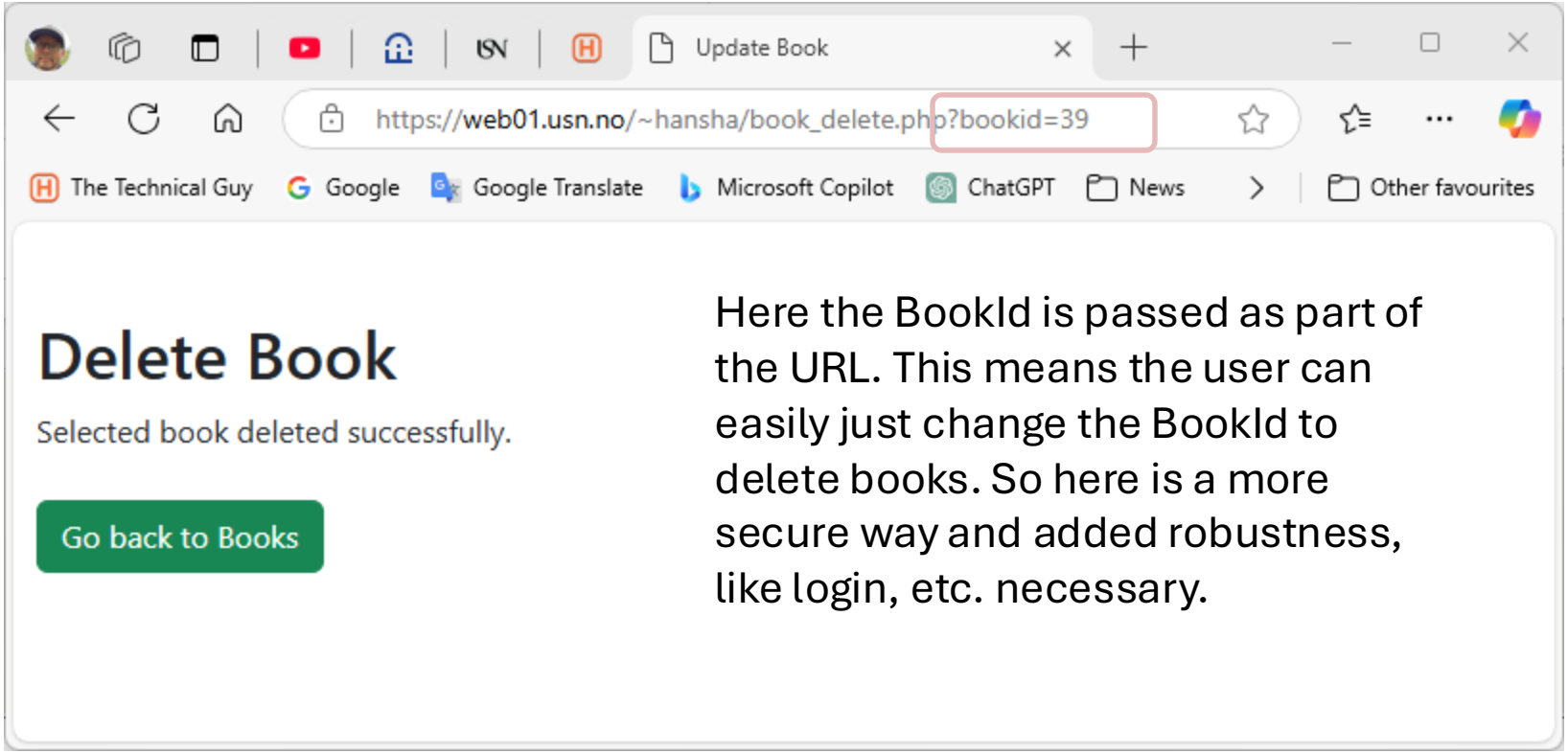
```
<!DOCTYPE html>
<html>
    <head>
        <title>Hello Friend!</title>
    </head>
    <body>

        <input type="text" value="Hans-Petter"><p></p>
        <input type="number" min="1" max="10"><p></p>
        <input type="password"><p></p>
        <input type="file"><p></p>

        <input type="checkbox"><p></p>
        <input type="radio"><p></p>
        <input type="range"><p></p>
        <input type="color"><p></p>

        <input type="date"><p></p>
        <input type="time"><p></p>
        <input type="datetime-local"><p></p>
        <input type="month"><p></p>

        <input type="button" value="Button"><p></p>
        <input type="submit">

    </body>
</html>
```



Make sure to use proper input types and not 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

Query String



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. Crete 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

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

Web: https://www.halvorsen.blog