https://www.halvorsen.blog

Create User page and Login page in PHP

Hans-Petter Halvorsen

Contents

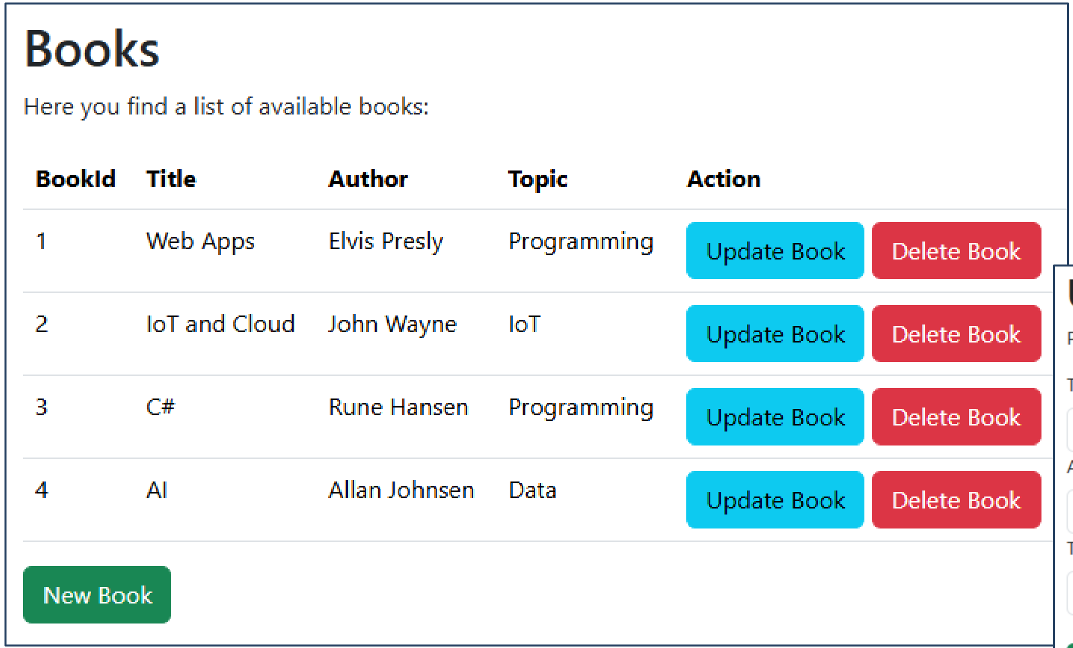
Introduction

Introduction

- This tutorial will demonstrate how to create basic login functionality in PHP.
- We show the principles on a Books/Library Application created earlier.
- The focus is to learn how to implement basic login functionality, so user experience, layout, robustness, etc. can be significantly improved compared to the simplified examples given here.

Books Web Application

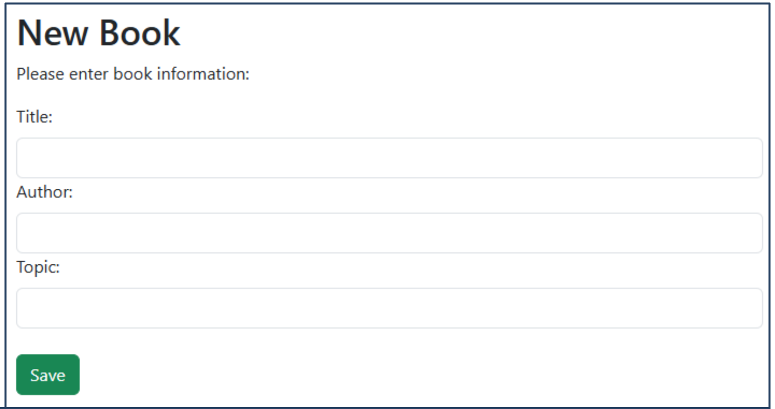
This is the existing PHP Application made. The focus was to create CRUD functionality, i.e., Create, Read, Update and Delete data in a MySQL database. The next step would be to add some authentication functionality.



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



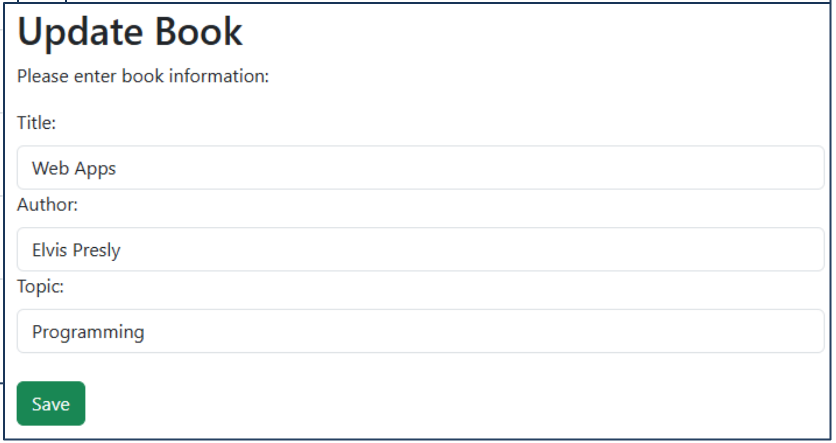
New Book

Please enter book information:

Title:

Author:

Topic:



Update Book

Please enter book information:

Title: Web Apps

Author: Elvis Presly

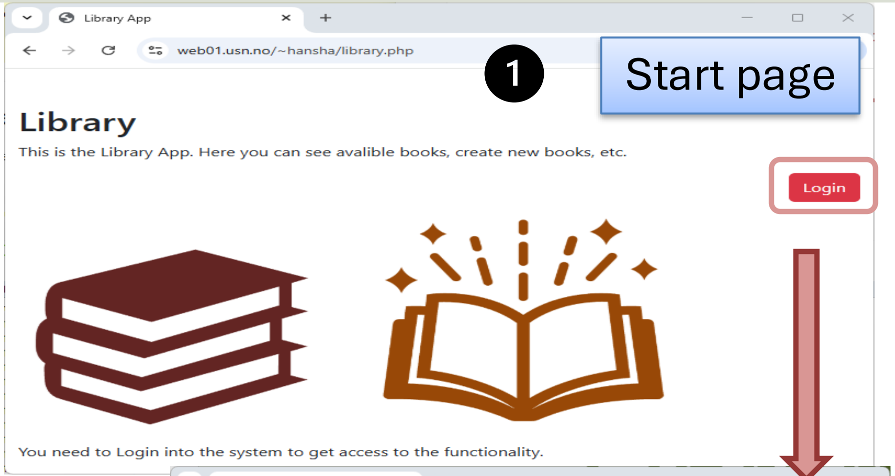
Topic: Programming

Create User and Login

We will update the Books WebApp with Login functionality. Here is a simple step by step procedure to do that:

1. Create a new Database Table with Username and Password information.
2. Crete a “NewUser” page that inserts UserName and Password information into the Database.
   - Here you can use the built-in “**password_hash**” PHP function.
3. Create a “Login” page 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)

New Library Application

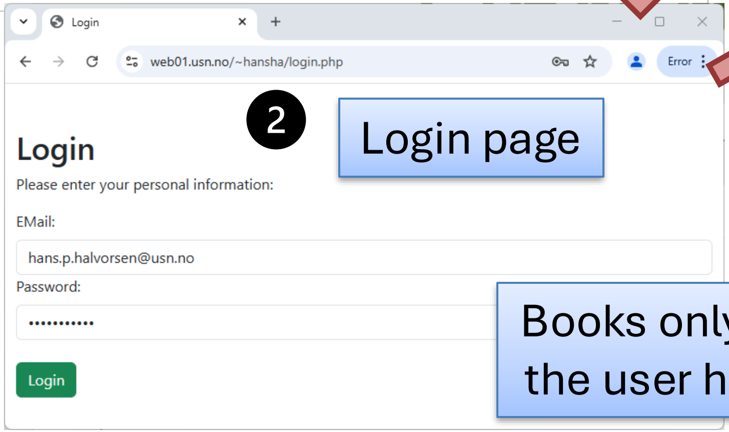


1 Start page

Library

This is the Library App. Here you can see avalible books, create new books, etc.

You need to Login into the system to get access to the functionality.



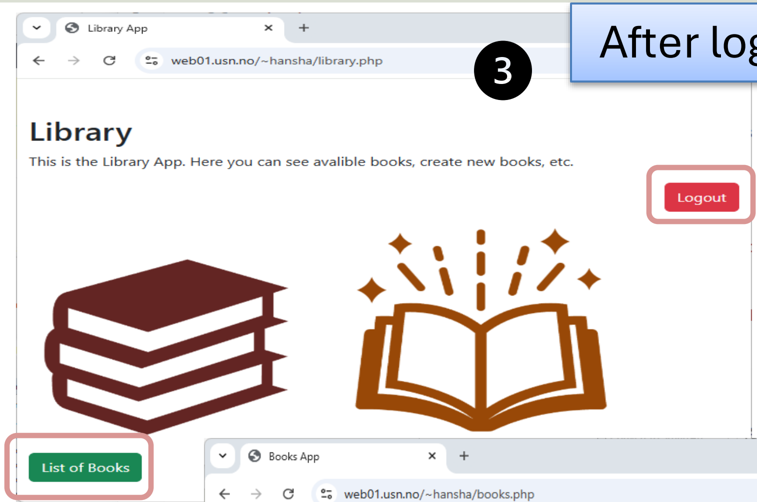
2 Login page

Login

Please enter your personal information:

EMail: hans.p.halvorsen@usn.no

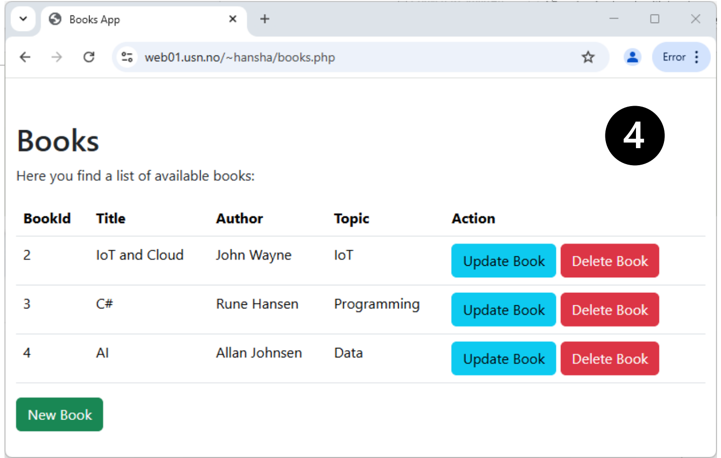
Password: ••••••••••••



3 After login

Library

This is the Library App. Here you can see avalible books, create new books, etc.



4

Books

Here you find a list of available books:

BookId	Title	Author	Topic	Action
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

Books only available if the user has logged in

Tools

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

Previous Resources

- HTML Tutorial – Here an introduction to HTML, CSS, JavaScript and Bootstrap is given.
- PHP Tutorial – Here an introduction to PHP is given.
- PHP and MySQL CRUD Web Application – Here a PHP CRUD Application from scratch is created (CRUD, Create, Read, Update and Delete information in a MySQL Database).

Security and Passwords

Hans-Petter Halvorsen

Security and Passwords

Here are some security and password related topics that we should know about.

- Authentication vs Authorization
- Encryption and Decrypting
- Hashing
- Salting
- 2 Factor Authentication
- Etc.

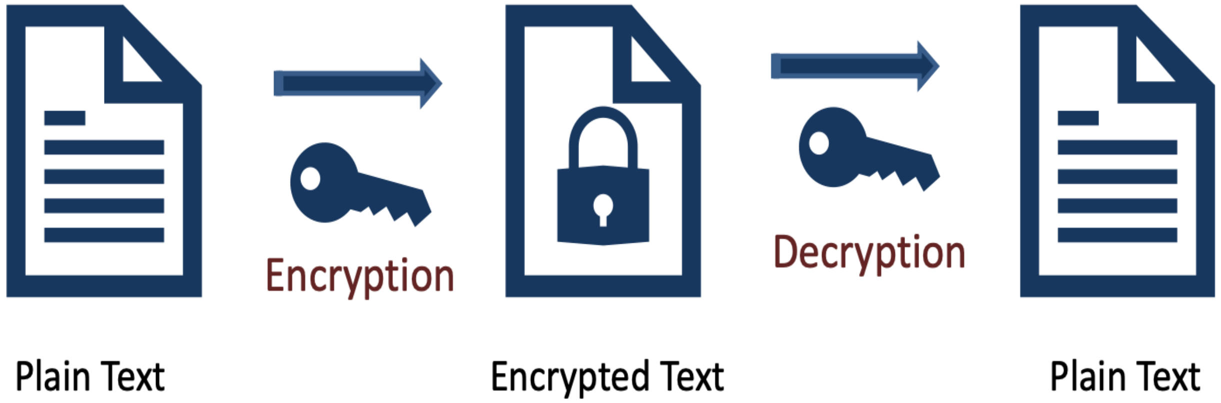
Authentication vs Authorization

- Authentication is a process to authenticate a user, that is, to verify that someone is who they say they are.
- Authorization is about determining a user's level of access and then granting access based on that level.

We will focus on Authentication in this tutorial.

Encryption and Decrypting

- Encryption is to transform information to make it unreadable by persons that should not have access to the information.
- You need a Key in to transform (decrypt) the information back to plain text.
- To encrypt data, you use an algorithm. Many different encryption algorithms do exist.



When should encryption be used?

- Encryption is a two-way function.
- You encrypt information with the intention of decrypting it later.
- Examples when to use encryption:
  - Protecting Files and Information on your Computer
  - Protecting your Cloud data
  - Transmitting Data between 2 Computers
  - Etc.
- Encryption is reversible if you have the key.
- It can be used for password protection, but a better approach is to use something called “Hashing”.

Hashing

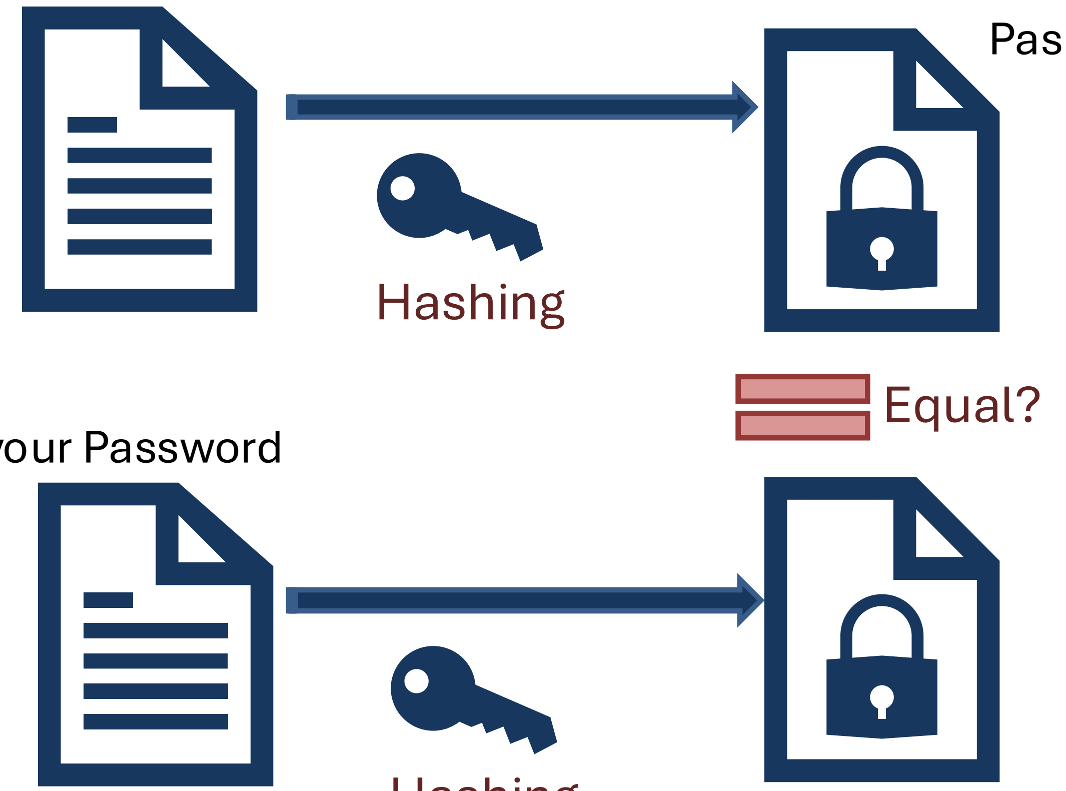
- Password hashing is a way to store passwords by transforming them into a fixed-length string of characters, which is not reversible.
- Hashing is a one-way function, while Encryption is a two-way function.
- Encryption is meant to protect data in transit, while Hashing is meant to verify that a file or piece of data hasn’t been altered—that it is authentic. In other words, it serves as a check-sum.
- Every hash value is unique.

Hashing

1. Create Password

Store Hashed Password in Database

Create a new Password in plain text



Compare the hashed password stored in the database with the hashed password the user enter when trying to login.

2. Login with your Password

Enter your Password in plain text

Hashed Password

Possible to hack the Hashed Password?

Password Table for System X

UserName	**HashedPassword**
Mike	4420d1918bbcf7
Bob	73fb51a0c9be7d
Peter	4420d1918bbcf7

If a Hacker gets access to this Database, he can see that Mike and Peter have the same password.
But he does not know the actual password.

Password	**HashedPassword**
tesla	4420d1918bbcf7
friendship	73fb51a0c9be7d
bicycle	7420e1618abcf6

Rainbow table

If the Hacker also has access to a so-called "**Rainbow table**" (which is essentially a pre-computed database of hashes), he may also be able to find the Password (as seen here).

If you have a complicated password and that you are not reusing your password several places, it is less likely that your password is in such a Rainbow table

Salting

- Salting is a technique typically used for Password Hashing.
- It is a unique value that can be added to the end of the password to create a different hash value.
- This means to identical passwords don’t end up with the same hashed value.
- The additional value is referred to as a “salt”.
- This is done to make it even more secure.
- Typically, the Hashing Algorithm uses a Random salt.
  - This prevents an attacker from seeing whether users have the same password.

Hashing with Salt

```
password = "Password123"
salt = "Tesla"

passwordHashed = HashPassword(password, salt);
```

Typically, Salting is built into the Hashing Algorithm, and it is changed every time

```
password = "Password123"

ph1 = HashPassword(password);
ph2 = HashPassword(password);
```

$ph1 \neq ph2$

This means if 2 different Users use the same Password, the Hashed Password will be different!

Hashing with Salt

Assume Mike and Peter use the same Password

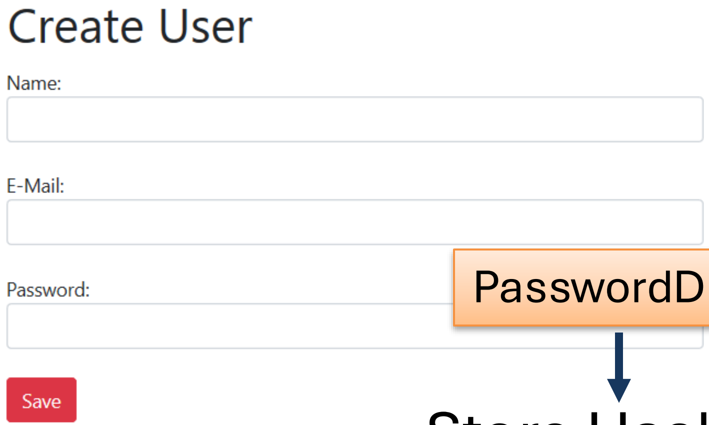
UserName	**Password**	**HashedPasswordwithSalt**
Mike	tesla	4420d1918bbcf7
Bob	friendship	73fb51a0c9be7d
Peter	tesla	4520d1818cbcf7

Different!

If a Hacker gets access to this Database, he cannot see that Mike and Peter have the same password.
Because a random Salt has made these 2 Hashed Passwords different!

Create User and Login

1



Create User

Name:

E-Mail:

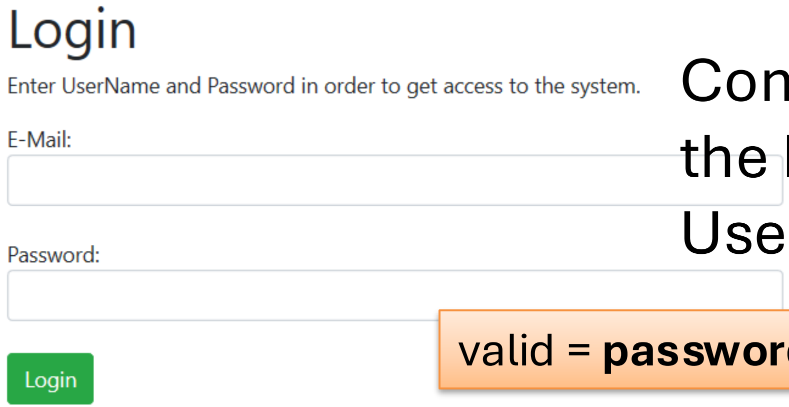
Password:

Information given by User

```
PasswordDB = password_hash(Password);
```

Store Hashed Password in the Database.

2



Login

Enter UserName and Password in order to get access to the system.

E-Mail:

Password:

Compare Hashed Password stored in the Database with Password given by User in Login Page.

```
valid = password_verify(PasswordDB, Password);
```

PHP Password Functions

1

```
$password = 'Password123';
$hashed_password = password_hash($password, PASSWORD_DEFAULT);
```

https://www.php.net/manual/en/function.password-hash.php

password_hash() has built-in salting

You can also choose between different types of hashing algorithms.

2

```
$verified = password_verify($password, $hashed_password);
```

https://www.php.net/manual/en/function.password-verify.php

Here, `$password` is entered by the user in the Login page and `$hashed_password` is the hashed password database stored in the database

2-Factor Authentication

- To make your authentication even better you can also consider implementing so-called 2-Factor Authentication.
- This can be done in different ways.
- The easiest way in PHP is to use email.
- You can use the built-in email() function in PHP
- The procedure is then to generate and send a verification code on email from your web application during the login process. Then the the user needs to enter the verification code received on email during the login process.

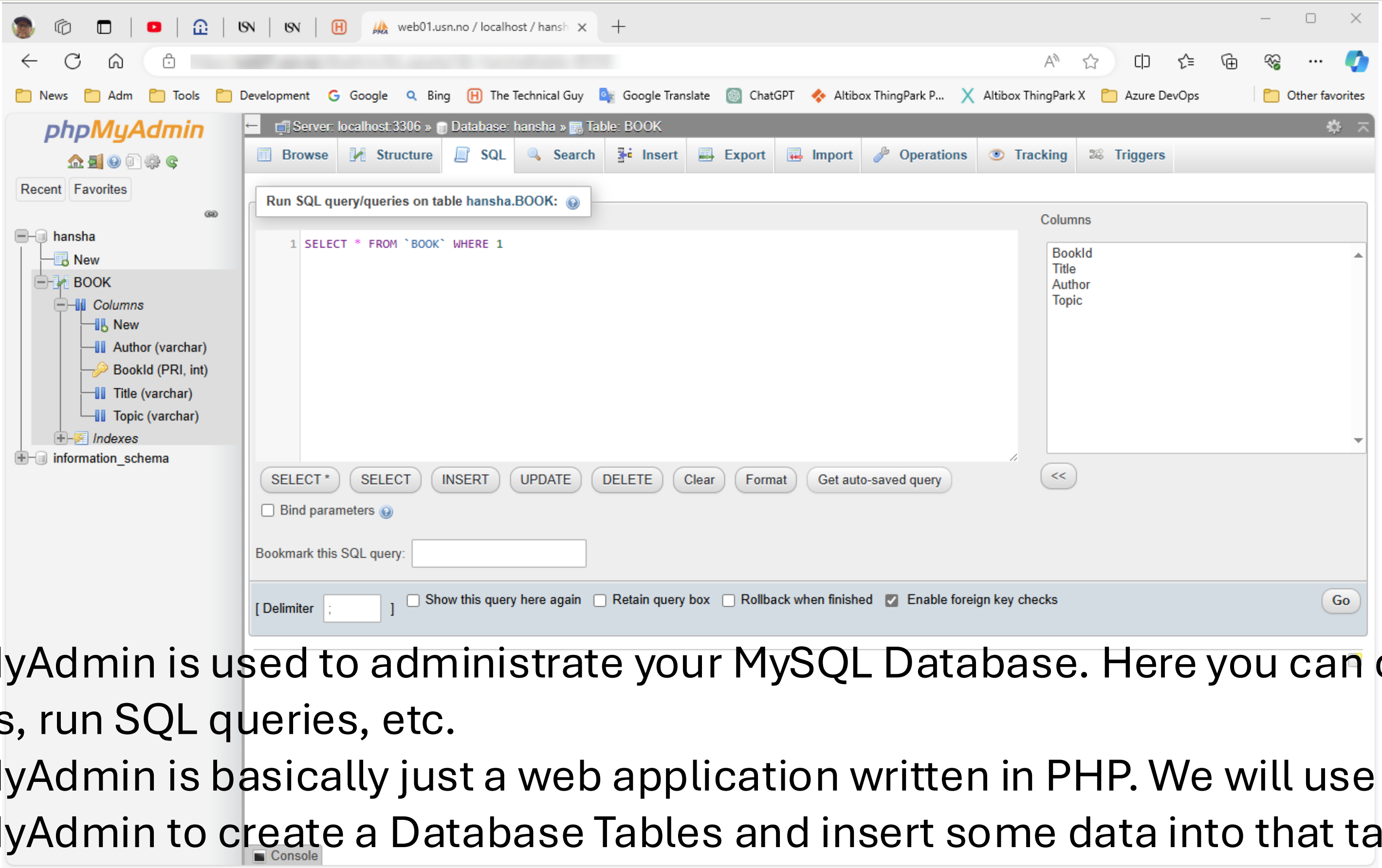
```
$verificationcode = strtoupper(uniqid());
```

```
mail($to, $subject, $message, $headers);
```

Database

Hans-Petter Halvorsen

phpMyAdmin



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

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

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

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

Library and Books

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

```
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');
```

User Information

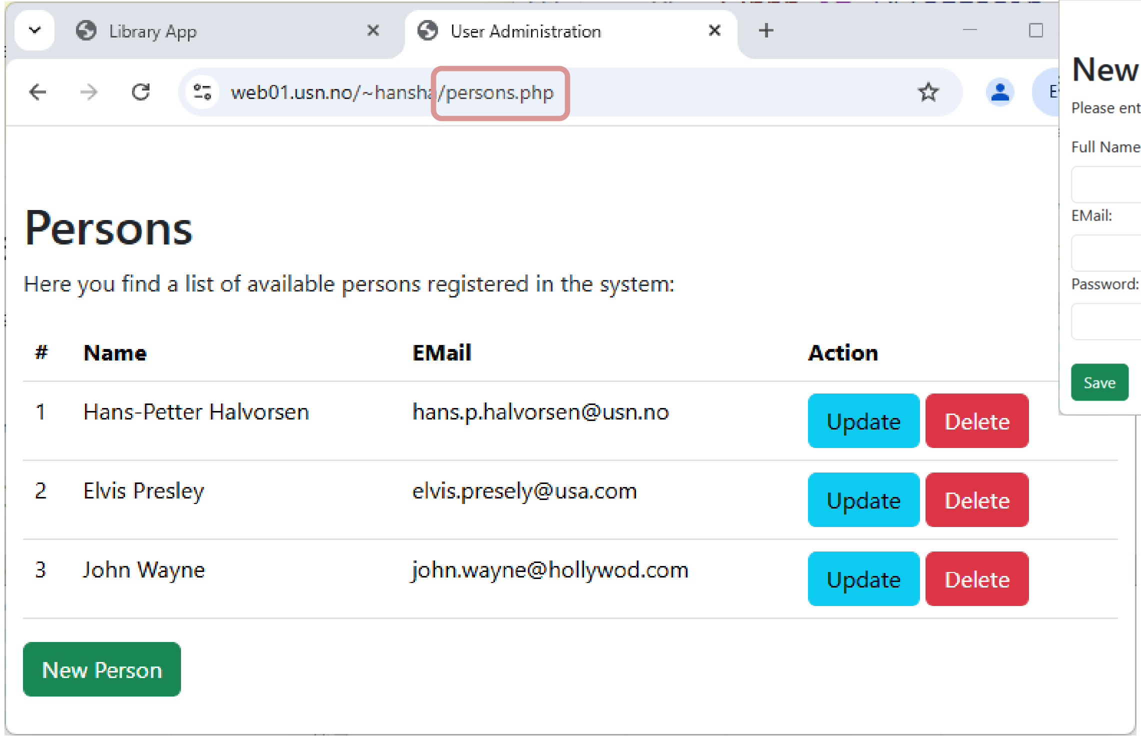
```
CREATE TABLE PERSON (
 PersonId int PRIMARY KEY AUTO_INCREMENT,
 FullName varchar(100) NOT NULL,
 EMail varchar(100) NOT NULL UNIQUE,
 Pwd varchar(100) NULL
);
```

```
insert into PERSON (FullName, EMail, Pwd)
values ('xxxxxx', 'xxxxxx', 'xxxxxx');
```

Create User

Hans-Petter Halvorsen

User Administration



Library App | User Administration

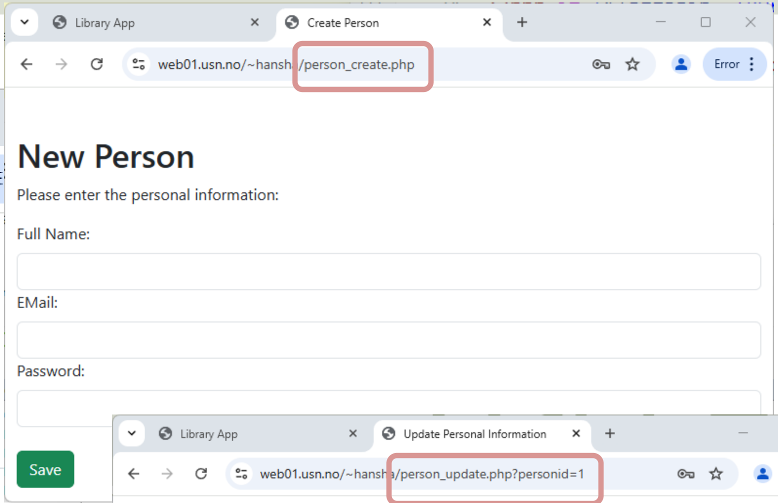
web01.usn.no/~hansha/persons.php

Persons

Here you find a list of available persons registered in the system:

#	Name	EMail	Action
1	Hans-Petter Halvorsen	hans.p.halvorsen@usn.no	Update Delete
2	Elvis Presley	elvis.presely@usa.com	Update Delete
3	John Wayne	john.wayne@hollywod.com	Update Delete

New Person



Library App | Create Person

web01.usn.no/~hansha/person_create.php

New Person

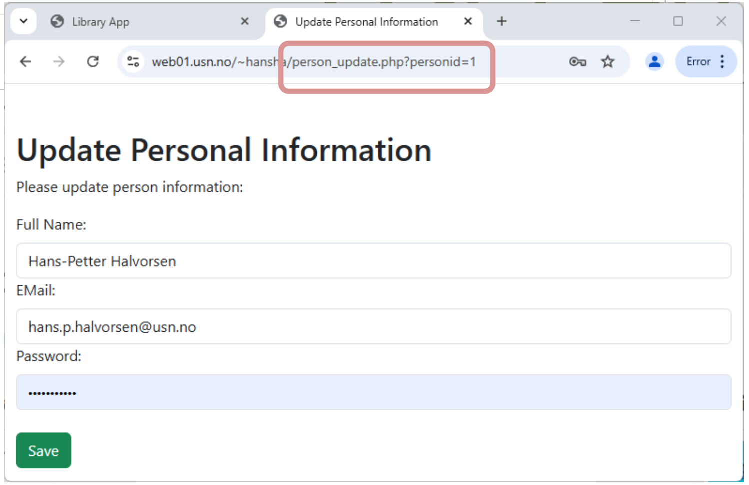
Please enter the personal information:

Full Name:

EMail:

Password:

Save



Library App | Update Personal Information

web01.usn.no/~hansha/person_update.php?personid=1

Update Personal Information

Please update person information:

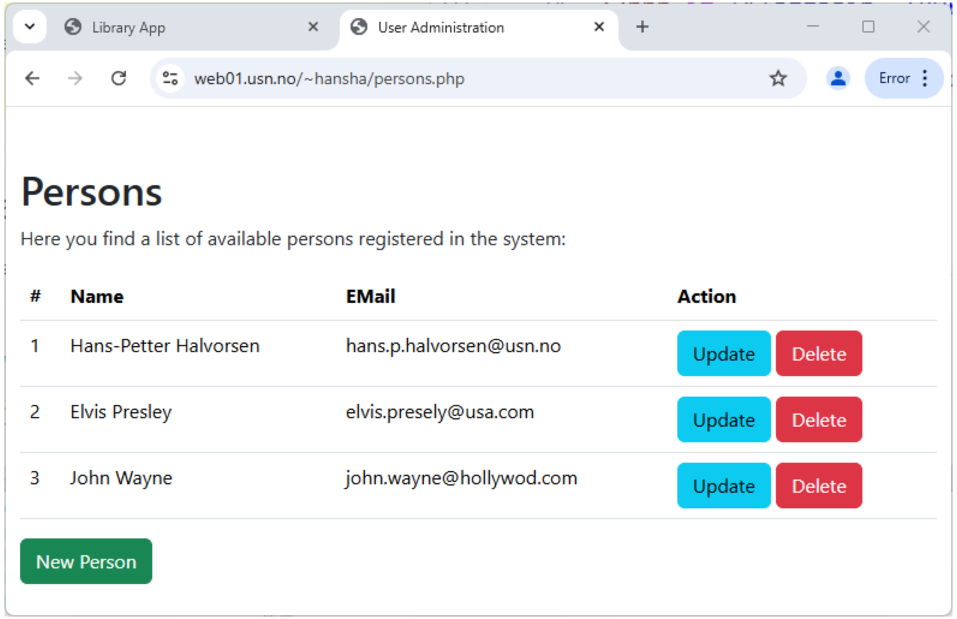
Full Name: Hans-Petter Halvorsen

EMail: hans.p.halvorsen@usn.no

Password: ••••••••••••

Save

Persons/Users

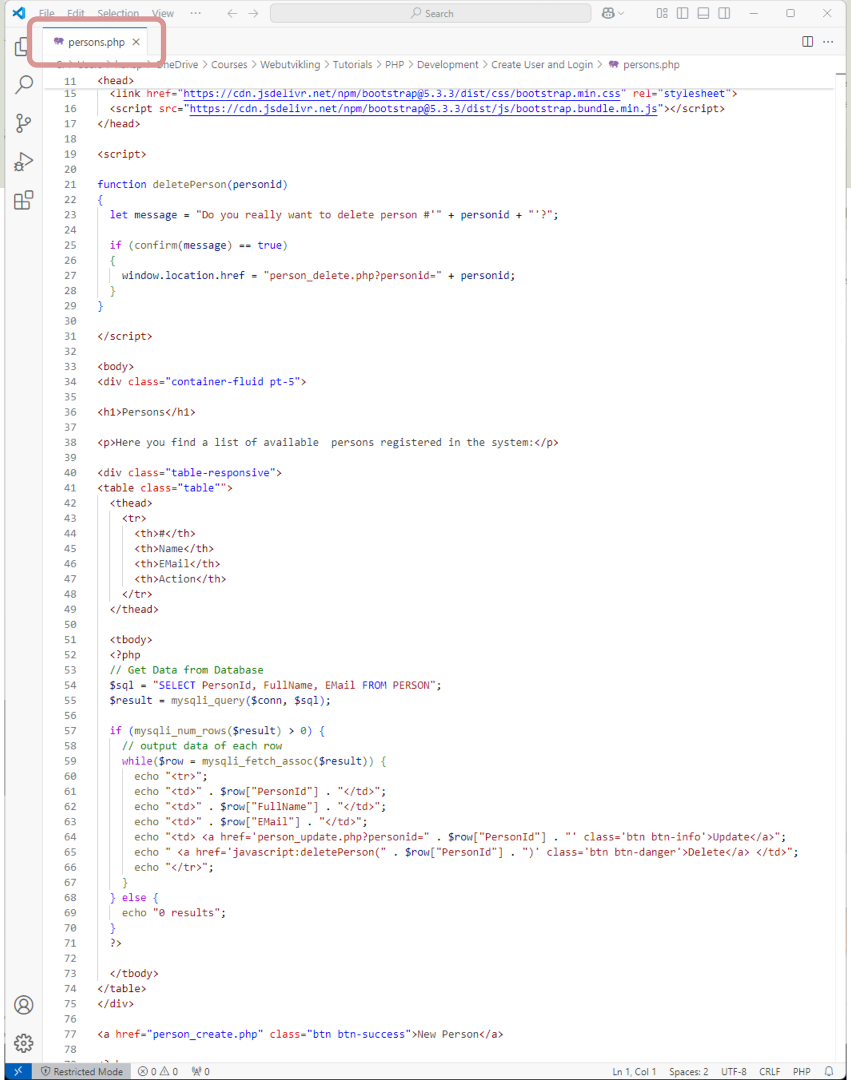


web01.usn.no/~hansha/persons.php

Persons

Here you find a list of available persons registered in the system:

#	Name	EMail	Action
1	Hans-Petter Halvorsen	hans.p.halvorsen@usn.no	Update Delete
2	Elvis Presley	elvis.presely@usa.com	Update Delete
3	John Wayne	john.wayne@hollywod.com	Update Delete

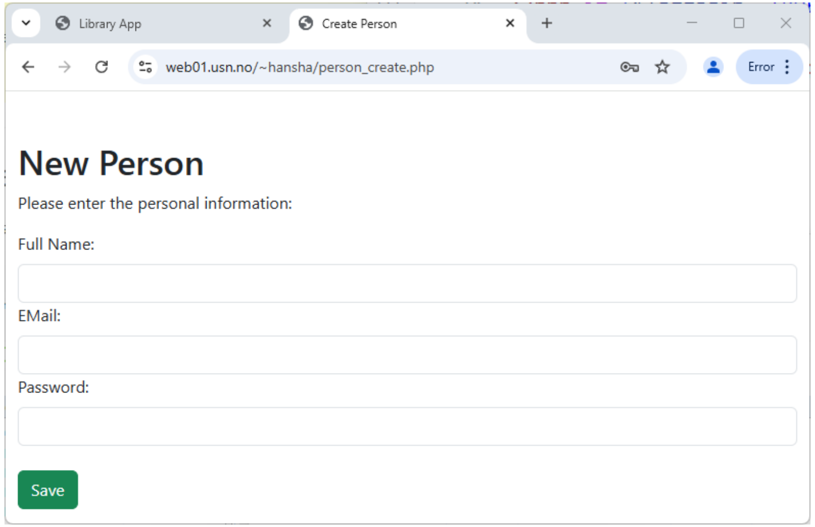


persons.php

```
<head>
  <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>

<script>

function deletePerson(personid)
{
  let message = "Do you really want to delete person #'" + personid + "'?";

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

</script>

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

<h1>Persons</h1>

<p>Here you find a list of available  persons registered in the system:</p>

<div class="table-responsive">
<table class="table"">
  <thead>
    <tr>
      <th>#</th>
      <th>Name</th>
      <th>EMail</th>
      <th>Action</th>
    </tr>
  </thead>

  <tbody>
  <?php
  // Get Data from Database
  $sql = "SELECT PersonId, FullName, EMail FROM PERSON";
  $result = mysqli_query($conn, $sql);

  if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
      echo "<tr>";
      echo "<td>" . $row["PersonId"] . "</td>";
      echo "<td>" . $row["FullName"] . "</td>";
      echo "<td>" . $row["EMail"] . "</td>";
      echo "<td> <a href='person_update.php?personid=" . $row["PersonId"] . "' class='btn btn-info'>Update</a>";
      echo " <a href='javascript:deletePerson(" . $row["PersonId"] . ")' class='btn btn-danger'>Delete</a> </td>";
      echo "</tr>";
    }
  } else {
    echo "0 results";
  }
  ?>

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

<a href="person_create.php" class="btn btn-success">New Person</a>
```

File
Edit
Selection
View
Search
OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Create User and Login > persons.php
Restricted Mode
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
PHP

New Person



New Person

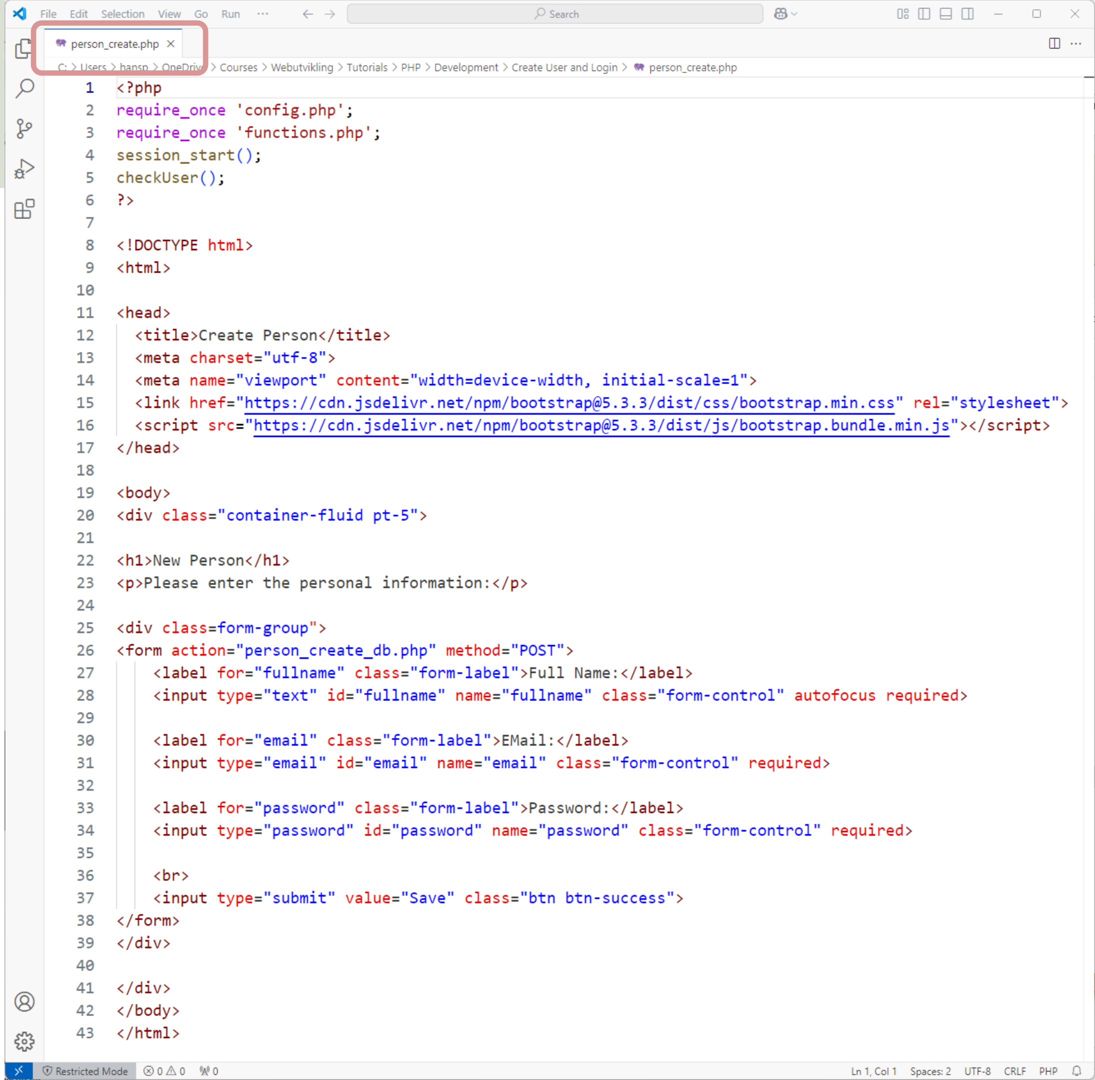
Please enter the personal information:

Full Name:

EMail:

Password:

Save

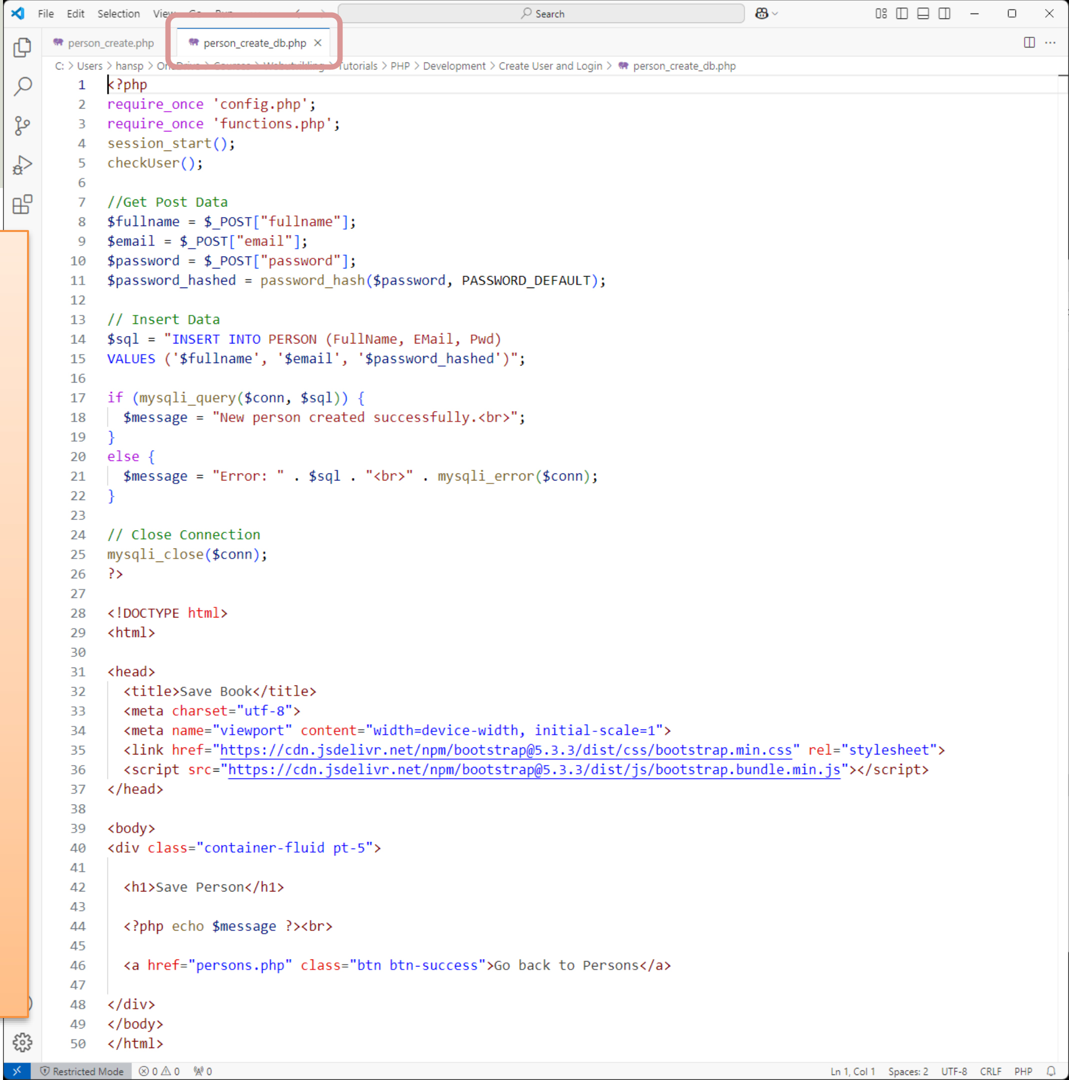


person_create.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Create User and Login > person_create.php
Restricted Mode
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF PHP

```
<?php
require_once 'config.php';
require_once 'functions.php';
session_start();
checkUser();
?>

<!DOCTYPE html>
<html>

<head>
  <title>Create Person</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>New Person</h1>
<p>Please enter the personal information:</p>

<div class=form-group">
<form action="person_create_db.php" method="POST">
    <label for="fullname" class="form-label">Full Name:</label>
    <input type="text" id="fullname" name="fullname" class="form-control" autofocus required>

    <label for="email" class="form-label">EMail:</label>
    <input type="email" id="email" name="email" class="form-control" required>

    <label for="password" class="form-label">Password:</label>
    <input type="password" id="password" name="password" class="form-control" required>

    <br>
    <input type="submit" value="Save" class="btn btn-success">
</form>
</div>

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

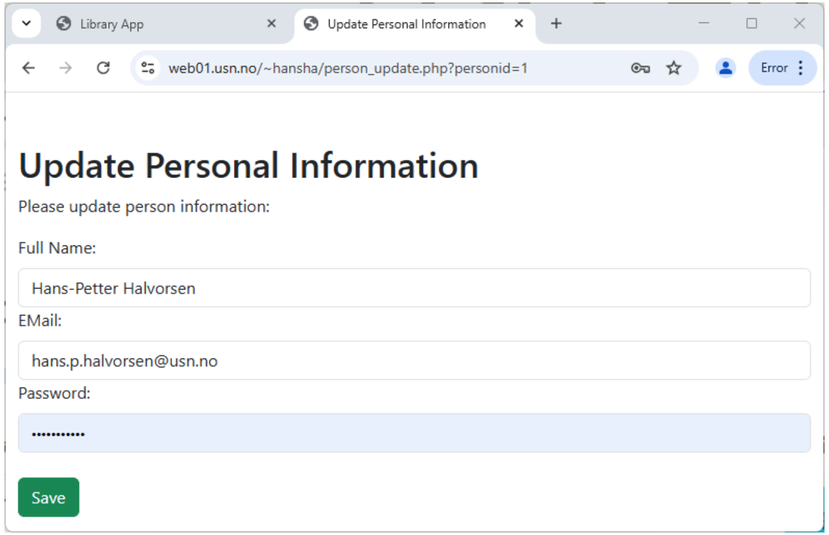
Save in DB

```
//Get Post Data
$fullname = $_POST["fullname"];
$email = $_POST["email"];
$password = $_POST["password"];
$password_hashed = password_hash($password, PASSWORD_DEFAULT);


// Insert Data
$sql = "INSERT INTO PERSON (FullName, EMail, Pwd)
VALUES ('$fullname', '$email', '$password_hashed')";


if (mysqli_query($conn, $sql)) {
  $message = "New person created successfully.<br>";
}
else {
  $message = "Error: " . $sql . "<br>" . mysqli_error($conn);
}
```



```
<?php
require_once 'config.php';
require_once 'functions.php';
session_start();
checkUser();

//Get Post Data
$fullname = $_POST["fullname"];
$email = $_POST["email"];
$password = $_POST["password"];
$password_hashed = password_hash($password, PASSWORD_DEFAULT);

// Insert Data
$sql = "INSERT INTO PERSON (FullName, EMail, Pwd)
VALUES ('$fullname', '$email', '$password_hashed')";

if (mysqli_query($conn, $sql)) {
  $message = "New person created successfully.<br>";
}
else {
  $message = "Error: " . $sql . "<br>" . mysqli_error($conn);
}

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

<!DOCTYPE html>
<html>

<head>
  <title>Save 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>Save Person</h1>

  <?php echo $message ?><br>

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

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

Update Person



Library App
Update Personal Information
web01.usn.no/~hansha/person_update.php?personid=1
Error

Update Personal Information

Please update person information:

Full Name:

Hans-Petter Halvorsen

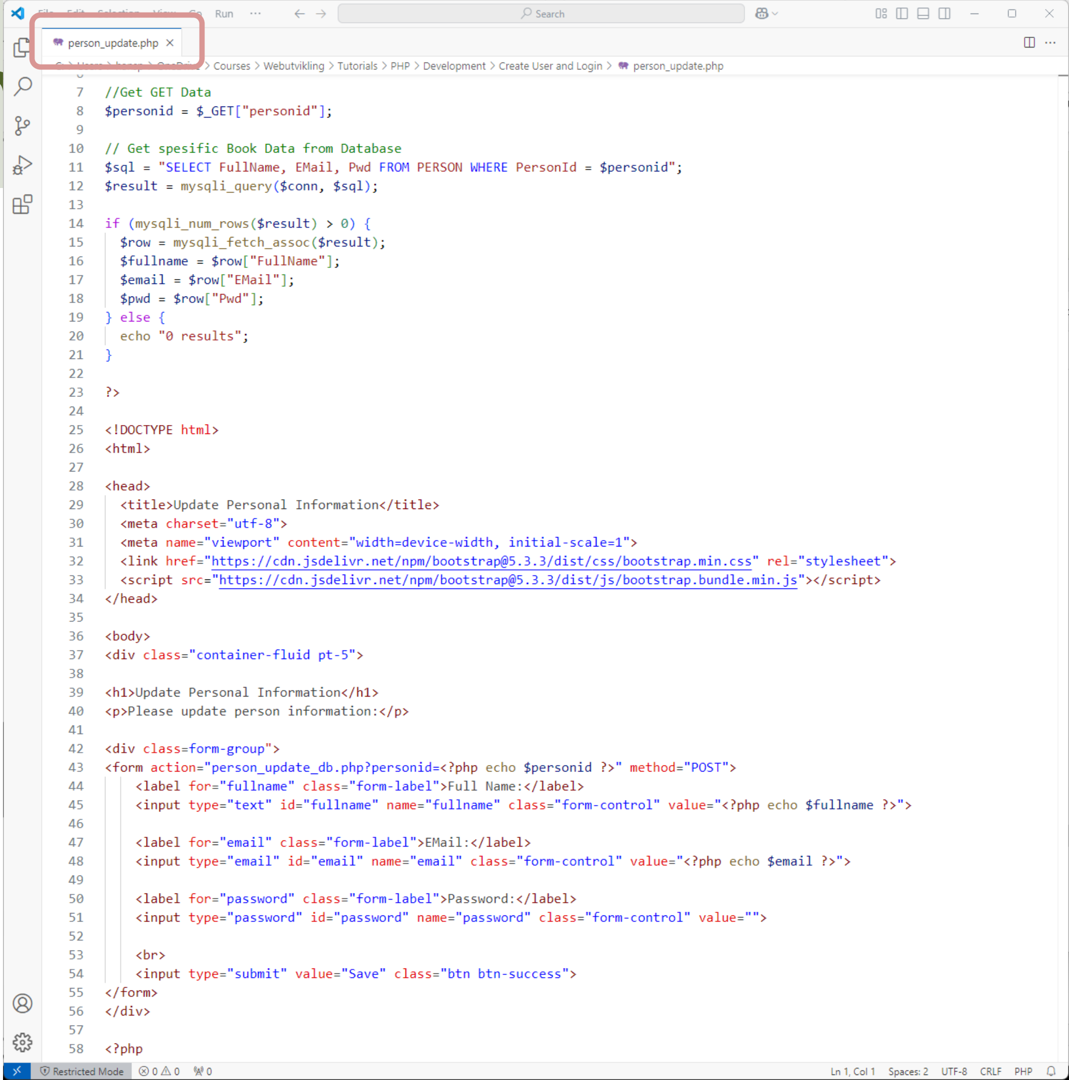
EMail:

hans.p.halvorsen@usn.no

Password:

•••••••••••

Save

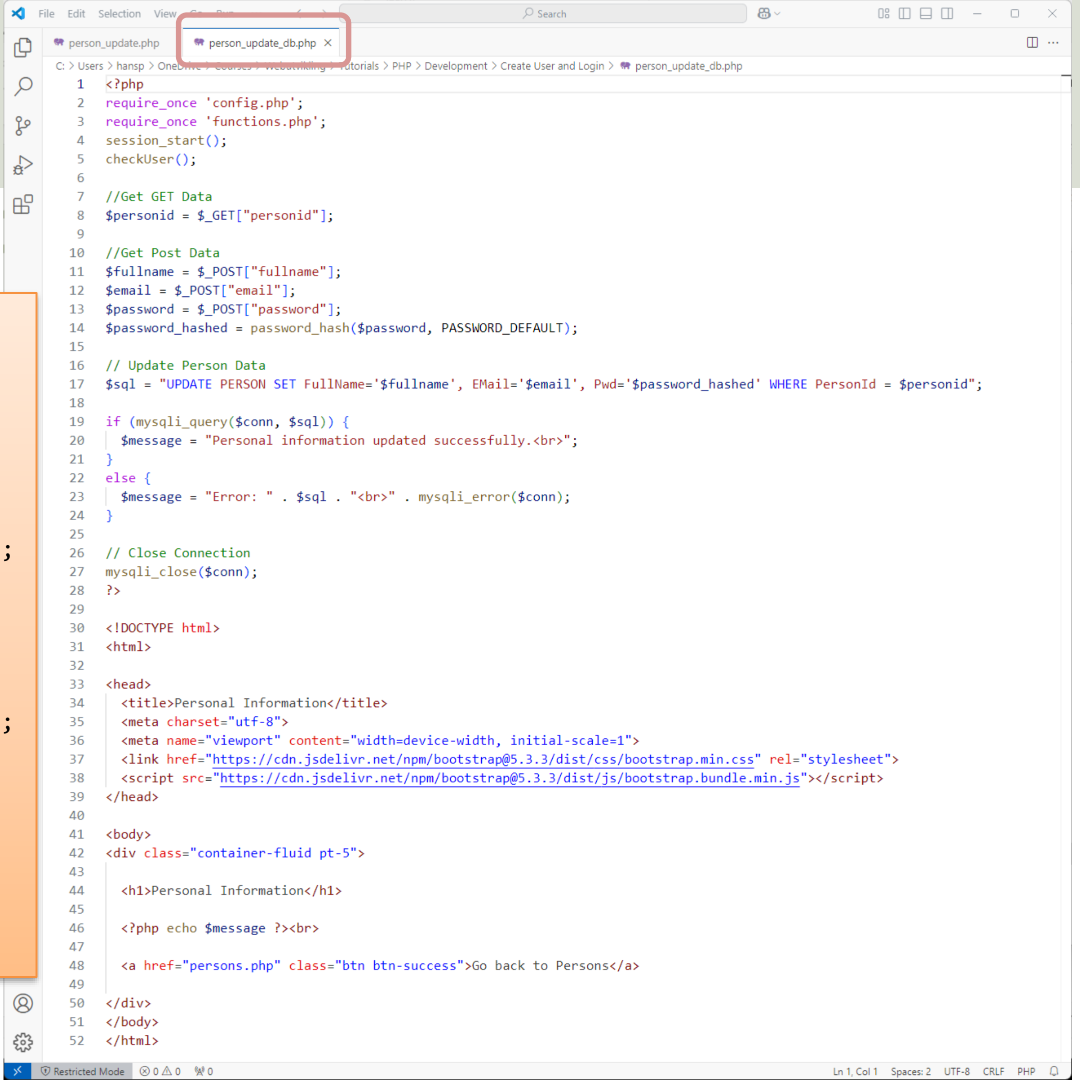


Run
Search
person_update.php
Courses > Webutvikling > Tutorials > PHP > Development > Create User and Login > person_update.php
Restricted Mode
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
PHP

```php
//Get GET Data
$personid = $_GET["personid"];

// Get spesific Book Data from Database
$sql = "SELECT FullName, EMail, Pwd FROM PERSON WHERE PersonId = $personid";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  $row = mysqli_fetch_assoc($result);
  $fullname = $row["FullName"];
  $email = $row["EMail"];
  $pwd = $row["Pwd"];
} else {
  echo "0 results";
}

?>

<!DOCTYPE html>
<html>

<head>
  <title>Update Personal Information</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>Update Personal Information</h1>
<p>Please update person information:</p>

<div class=form-group">
<form action="person_update_db.php?personid=<?php echo $personid ?>" method="POST">
    <label for="fullname" class="form-label">Full Name:</label>
    <input type="text" id="fullname" name="fullname" class="form-control" value="<?php echo $fullname ?>">

    <label for="email" class="form-label">EMail:</label>
    <input type="email" id="email" name="email" class="form-control" value="<?php echo $email ?>">

    <label for="password" class="form-label">Password:</label>
    <input type="password" id="password" name="password" class="form-control" value="">

    <br>
    <input type="submit" value="Save" class="btn btn-success">
</form>
</div>

<?php
```

Update DB

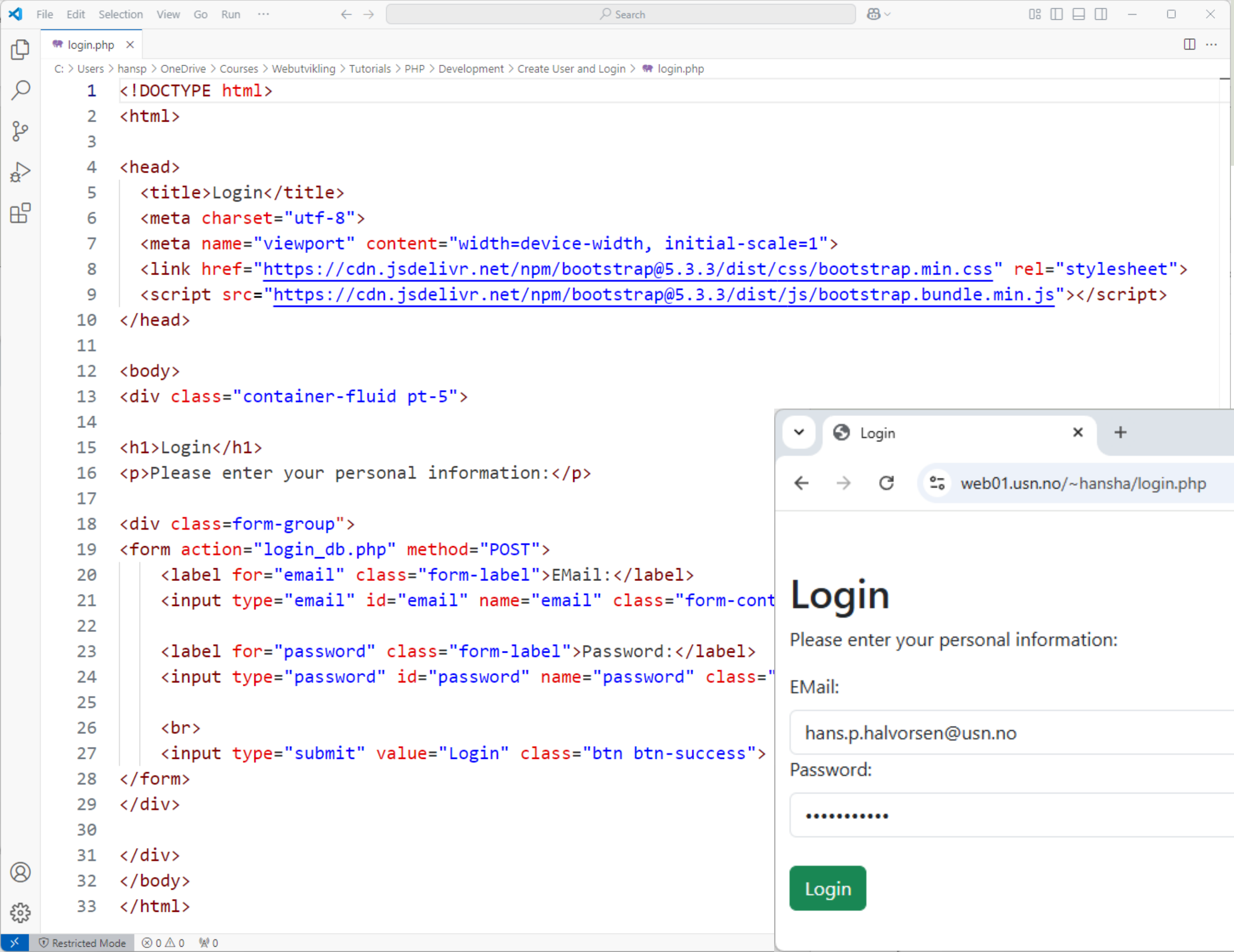
```php
<?php
..
//Get GET Data
$personid = $_GET["personid"];
//Get Post Data
$fullname = $_POST["fullname"];
$email = $_POST["email"];
$password = $_POST["password"];
$password_hashed = password_hash($password, PASSWORD_DEFAULT);
// Update Person Data
$sql = "UPDATE PERSON SET FullName='$fullname',
EMail='$email', Pwd='$password_hashed' WHERE PersonId =
$personid";
if (mysqli_query($conn, $sql)) {
  $message = "Personal information updated successfully.<br>";
}
else {
  $message = "Error: " . $sql . "<br>" . mysqli_error($conn);
}

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


File
Edit
Selection
View
Search
person_update.php
person_update_db.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Create User and Login > person_update_db.php
Restricted Mode
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
PHP

```php
<?php
require_once 'config.php';
require_once 'functions.php';
session_start();
checkUser();

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

//Get Post Data
$fullname = $_POST["fullname"];
$email = $_POST["email"];
$password = $_POST["password"];
$password_hashed = password_hash($password, PASSWORD_DEFAULT);

// Update Person Data
$sql = "UPDATE PERSON SET FullName='$fullname', EMail='$email', Pwd='$password_hashed' WHERE PersonId = $personid";

if (mysqli_query($conn, $sql)) {
  $message = "Personal information updated successfully.<br>";
}
else {
  $message = "Error: " . $sql . "<br>" . mysqli_error($conn);
}

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

<!DOCTYPE html>
<html>

<head>
  <title>Personal Information</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>Personal Information</h1>

  <?php echo $message ?><br>

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

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

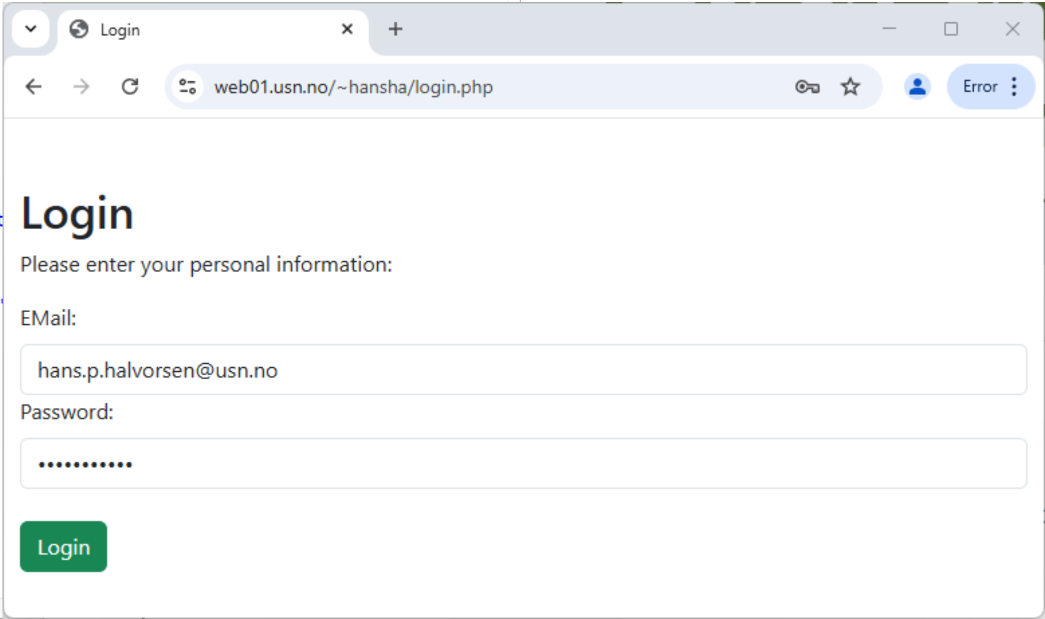
Login

Login



File Edit Selection View Go Run
login.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Create User and Login > login.php
Restricted Mode

```
<!DOCTYPE html>
<html>

<head>
  <title>Login</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>Login</h1>
<p>Please enter your personal information:</p>

<div class=form-group">
<form action="login_db.php" method="POST">
    <label for="email" class="form-label">EMail:</label>
    <input type="email" id="email" name="email" class="form-cont

    <label for="password" class="form-label">Password:</label>
    <input type="password" id="password" name="password" class="

    <br>
    <input type="submit" value="Login" class="btn btn-success">
</form>
</div>

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



Login

Please enter your personal information:

EMail:

hans.p.halvorsen@usn.no

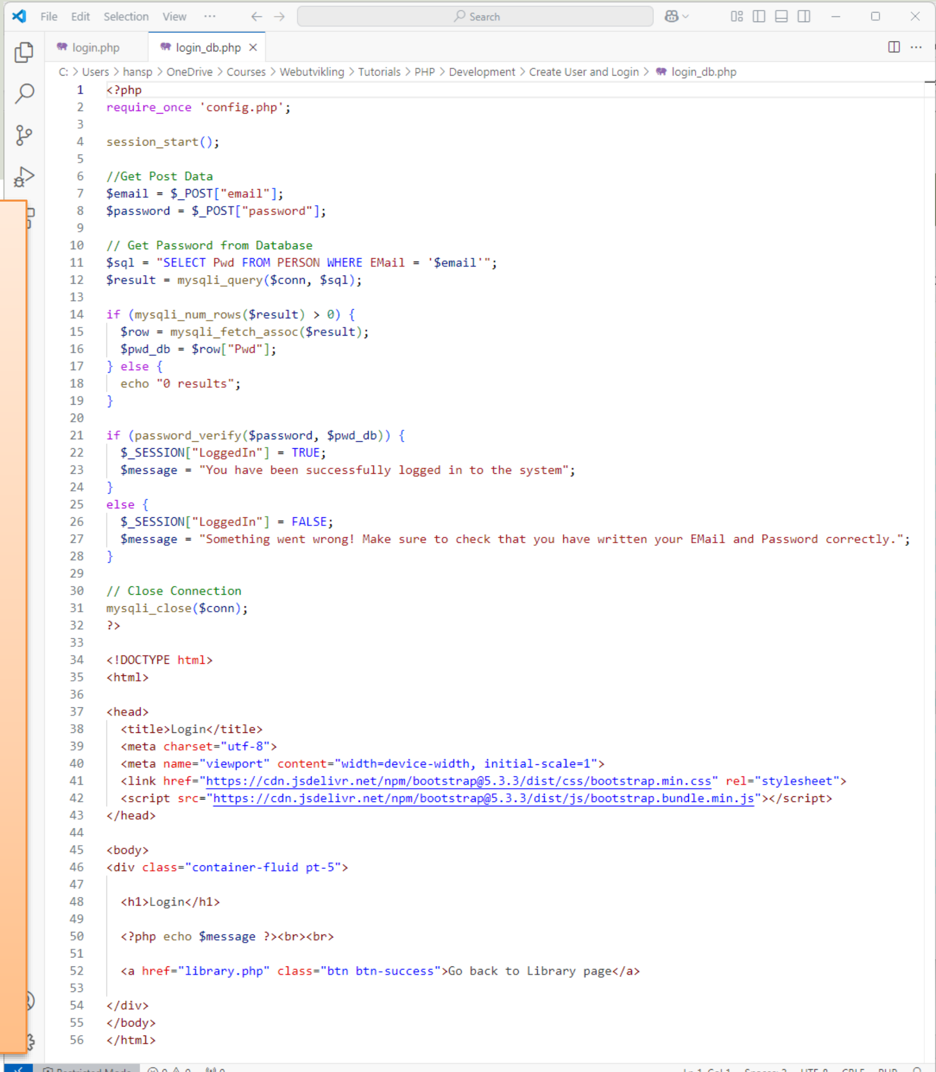
Password:

•••••••••••

Login

Check Login

```php
//Get Post Data
$email = $_POST["email"];
$password = $_POST["password"];

// Get Password from Database
$sql = "SELECT Pwd FROM PERSON WHERE EMail = '$email'";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  $row = mysqli_fetch_assoc($result);
  $pwd_db = $row["Pwd"];
} else {
  echo "0 results";
}

if (password_verify($password, $pwd_db)) {
  $_SESSION["LoggedIn"] = TRUE;
  $message = "You have been successfully logged in ..";
}
else {
  $_SESSION["LoggedIn"] = FALSE;
  $message = "Something went wrong! Make sure to check ..";
}
```


File Edit Selection View ···
Search
login.php
login_db.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Create User and Login > login_db.php

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

session_start();

//Get Post Data
$email = $_POST["email"];
$password = $_POST["password"];

// Get Password from Database
$sql = "SELECT Pwd FROM PERSON WHERE EMail = '$email'";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  $row = mysqli_fetch_assoc($result);
  $pwd_db = $row["Pwd"];
} else {
  echo "0 results";
}

if (password_verify($password, $pwd_db)) {
  $_SESSION["LoggedIn"] = TRUE;
  $message = "You have been successfully logged in to the system";
}
else {
  $_SESSION["LoggedIn"] = FALSE;
  $message = "Something went wrong! Make sure to check that you have written your EMail and Password correctly.";
}

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

<!DOCTYPE html>
<html>

<head>
  <title>Login</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>Login</h1>

  <?php echo $message ?><br><br>

  <a href="library.php" class="btn btn-success">Go back to Library page</a>

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

Using Session variables

- We need to store information whether the User is logged in or not when the user enters the different pages.
- We can use Session variables to share that information between multiple web pages.

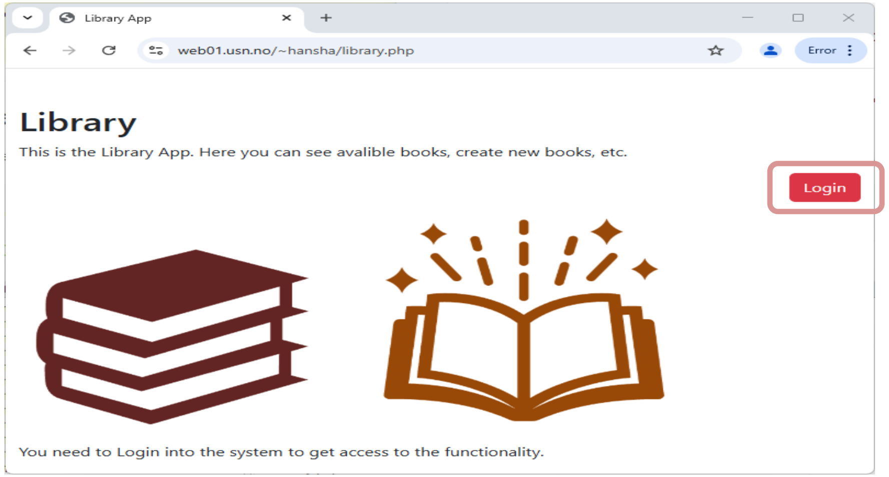
```
session_start();
..
$_SESSION["LoggedIn"] = TRUE;
```

Check if User is Logged in

Check if User is Logged in

- We don’t want the user to get access to different web pages if he has not logged in.
- 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)

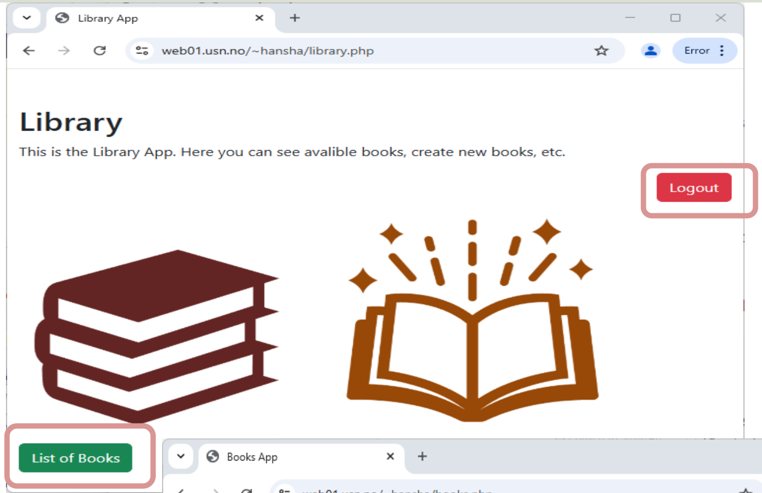
Check if User is Logged in



Library

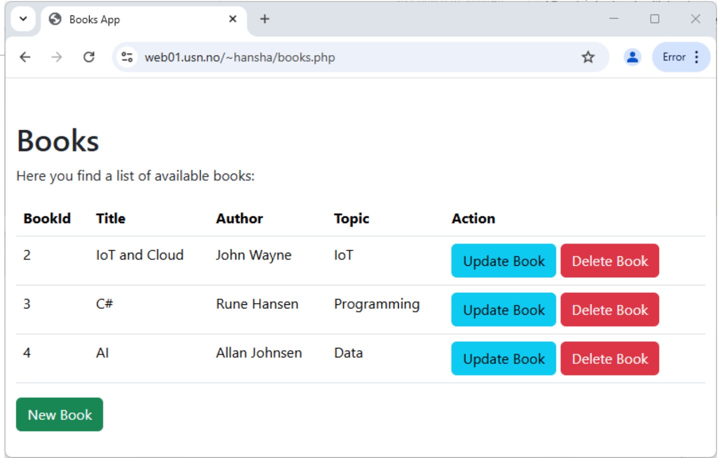
This is the Library App. Here you can see avalible books, create new books, etc.

You need to Login into the system to get access to the functionality.



Library

This is the Library App. Here you can see avalible books, create new books, etc.



Books

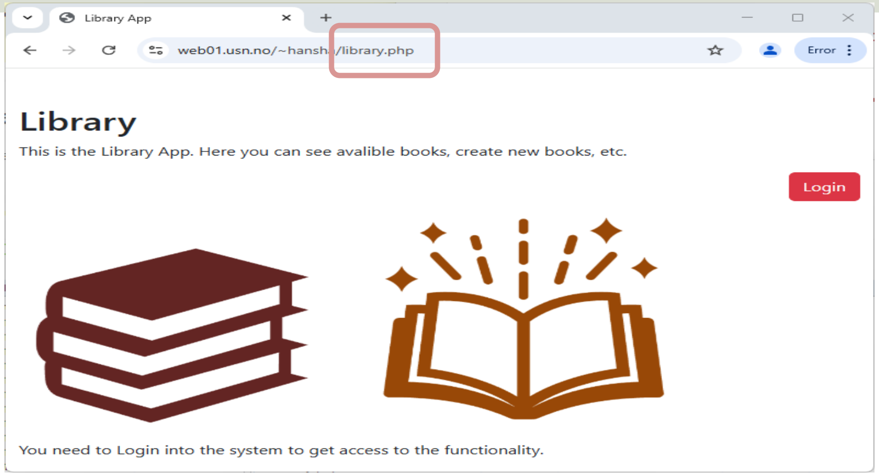
Here you find a list of available books:

BookId	Title	Author	Topic	Action
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

New Book

The Library page will have different appearance/functionality depending on the user is logged in or not.

Library page

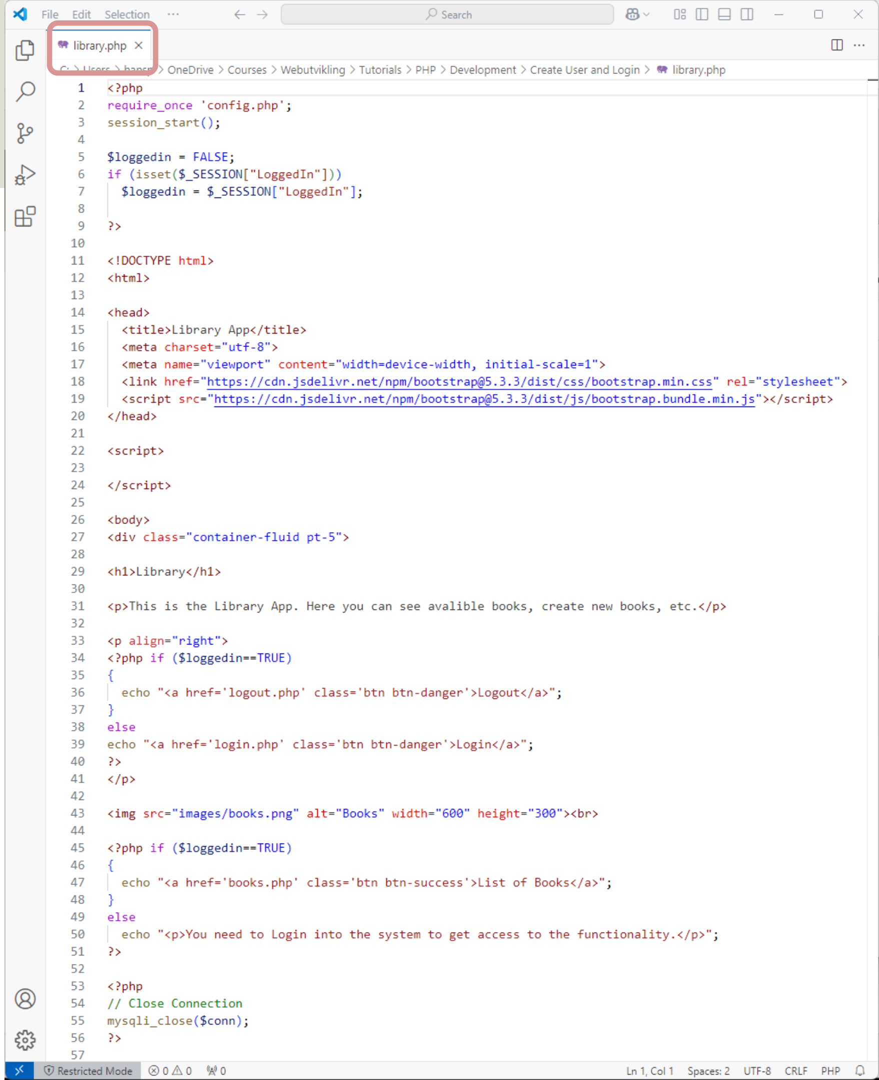


Library

This is the Library App. Here you can see avalible books, create new books, etc.

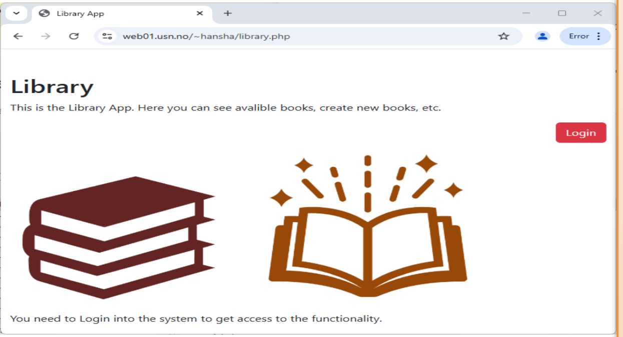
You need to Login into the system to get access to the functionality.

```php
<?php
require_once 'config.php';
session_start();

$loggedin = FALSE;
if (isset($_SESSION["LoggedIn"]))
  $loggedin = $_SESSION["LoggedIn"];

?>
..
```


File Edit Selection ...
Search
library.php
C: > Users > hansh > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Create User and Login > library.php
Restricted Mode
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF PHP

```php
<?php
require_once 'config.php';
session_start();

$loggedin = FALSE;
if (isset($_SESSION["LoggedIn"]))
  $loggedin = $_SESSION["LoggedIn"];

?>

<!DOCTYPE html>
<html>

<head>
  <title>Library App</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>

<script>

</script>

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

<h1>Library</h1>

<p>This is the Library App. Here you can see avalible books, create new books, etc.</p>

<p align="right">
<?php if ($loggedin==TRUE)
{
  echo "<a href='logout.php' class='btn btn-danger'>Logout</a>";
}
else
echo "<a href='login.php' class='btn btn-danger'>Login</a>";
?>
</p>

<img src="images/books.png" alt="Books" width="600" height="300"><br>

<?php if ($loggedin==TRUE)
{
  echo "<a href='books.php' class='btn btn-success'>List of Books</a>";
}
else
  echo "<p>You need to Login into the system to get access to the functionality.</p>";
?>

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

Library page Code cont.

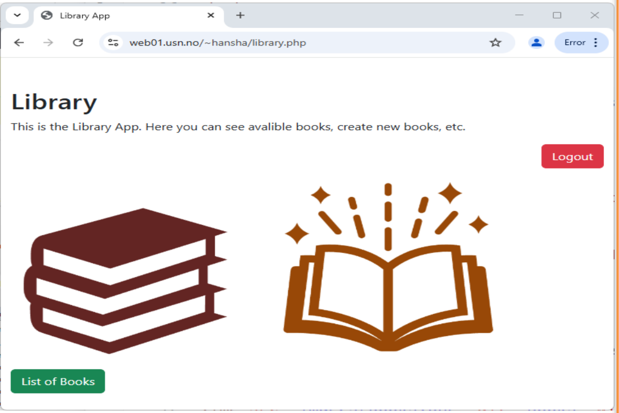


Library

This is the Library App. Here you can see avalible books, create new books, etc.

Login

You need to Login into the system to get access to the functionality.



Library

This is the Library App. Here you can see avalible books, create new books, etc.

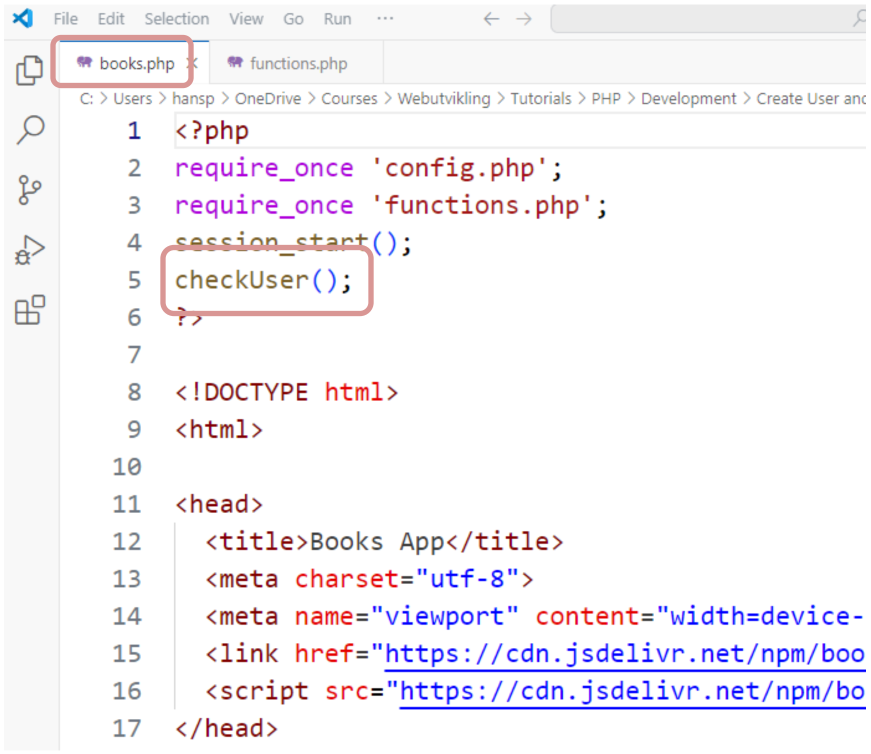
Logout

List of Books

```
..
<p align="right">
<?php if ($loggedin==TRUE)
{
  echo "<a href='logout.php' class='btn btn-danger'>Logout</a>";
}
else
echo "<a href='login.php' class='btn btn-danger'>Login</a>";
?>
</p>

<img src="images/books.png" alt="Books" width="600" height="300"><br>

<?php if ($loggedin==TRUE)
{
  echo "<a href='books.php' class='btn btn-success'>List of Books</a>";
}
else
  echo "<p>You need to Login into the system to get access to the
functionality.</p>";
?>
..
```

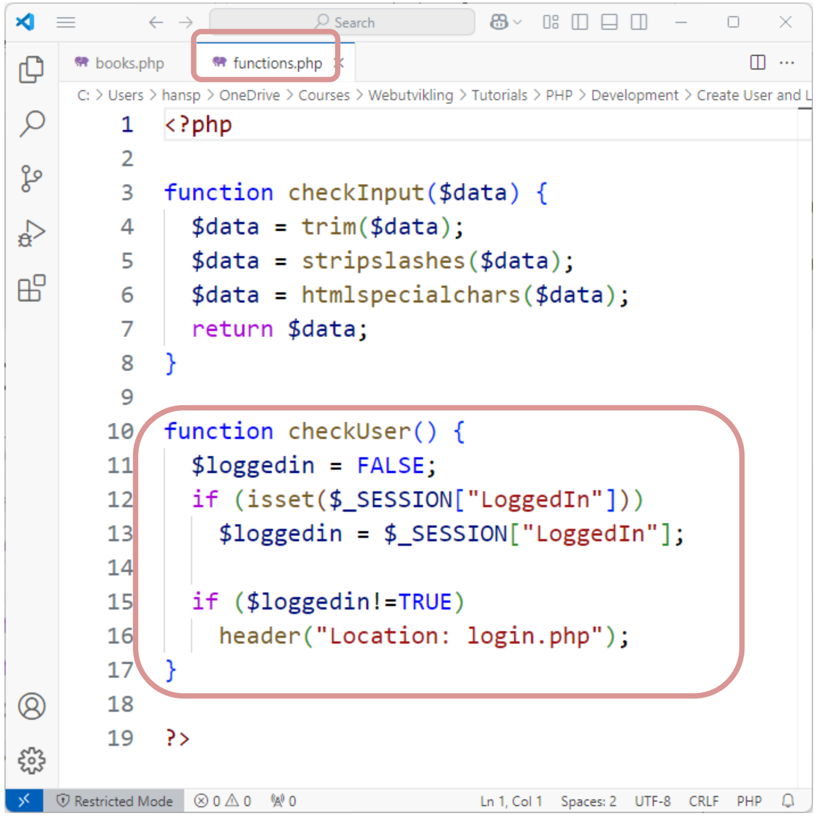
Books page

The same check needs to be done in the different Books pages, etc.



File Edit Selection View Go Run ···
books.php
functions.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Create User anc

```
<?php
require_once 'config.php';
require_once 'functions.php';
session_start();
checkUser();
?>

<!DOCTYPE html>
<html>

<head>
  <title>Books App</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-
  <link href="https://cdn.jsdelivr.net/npm/boo
  <script src="https://cdn.jsdelivr.net/npm/bo
</head>
```



Search
books.php
functions.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > Create User and L
Restricted Mode
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF PHP

```
<?php

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

function checkUser() {
  $loggedin = FALSE;
  if (isset($_SESSION["LoggedIn"]))
    $loggedin = $_SESSION["LoggedIn"];

  if ($loggedin!=TRUE)
    header("Location: login.php");
}

?>
```

2-Factor Authentication

Hans-Petter Halvorsen

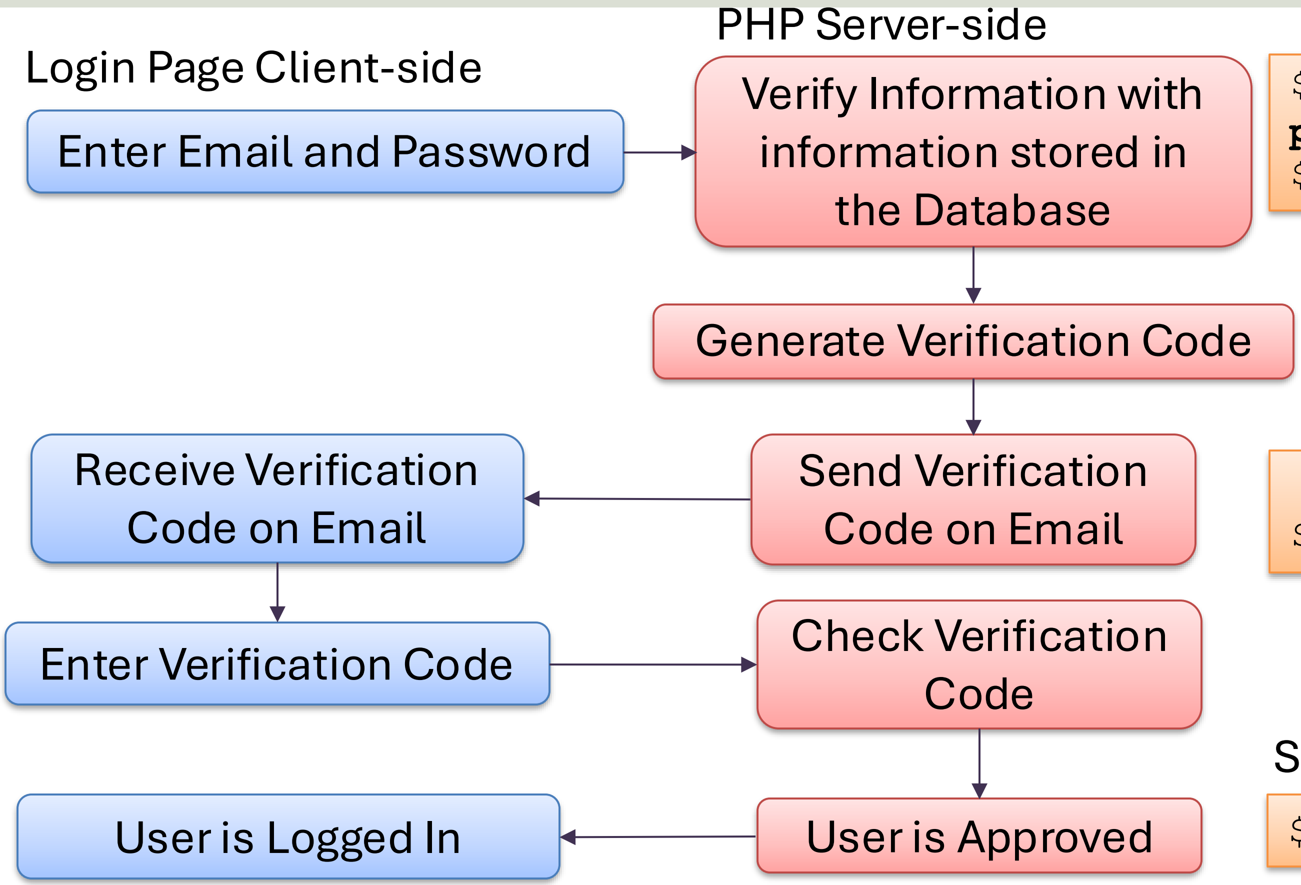
2-Factor Authentication

- To make your authentication even better you can also consider implementing so-called 2-Factor Authentication.
- This can be done in different ways.
- The easiest way in PHP is to use email.
- You can use the built-in email() function in PHP
- The procedure is then to generate and send a verification code on email from your web application during the login process. Then the the user needs to enter the verification code received on email during the login process.

```
$verificationcode = strtoupper(uniqid());
```

```
mail($to, $subject, $message, $headers);
```

2-Factor Authentication


PHP Server-side
Login Page Client-side
Enter Email and Password
Verify Information with information stored in the Database
Generate Verification Code
Send Verification Code on Email
Receive Verification Code on Email
Enter Verification Code
Check Verification Code
User is Approved
User is Logged In

```
$verified =
password_verify($password,
$hashed_password);
```

```
$verificationcode =
strtoupper(uniqid());
```

```
 mail($to, $subject,
$message, $headers);
```

Set Session variable, e.g.,:

```
$_SESSION["LOGIN"]=TRUE
```

Authorization

Hans-Petter Halvorsen

Authentication vs Authorization

- Authentication is a process to authenticate a user, that is, to verify that someone is who they say they are.
- Authorization is about determining a user's level of access and then granting access based on that level.

So far, we have focused on Authentication in this tutorial.
How can we implement Authorization?

Authorization

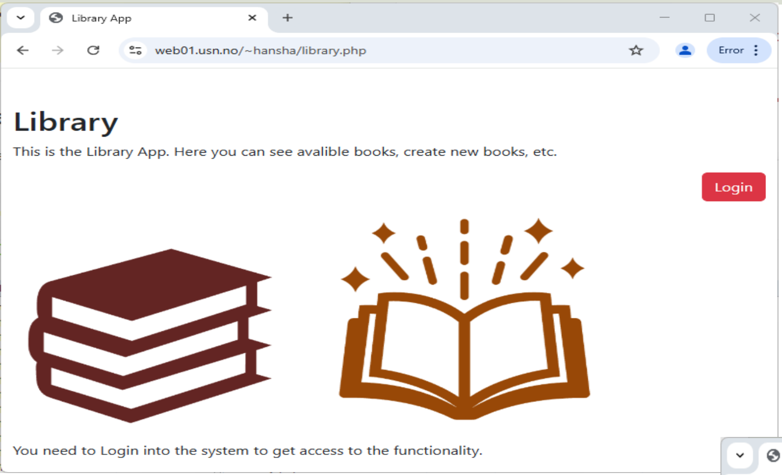
In out “Library” application we may want to have different types of users, examples:

- Administrators
  - They can add new Users, edit and delete Users.
- Librarians
  - They can create new Books, edit or delete books.
  - They can see information regarding book rentals, ertc.
- End users
  - They have access to book information, but they cannot create new books, edit or delete books
  - They can rent books, see if books are available or not

=>This can be implemented in different ways, but typically you need to start to add one or more new tables or columns in the database to handle this in your application. This will not be part of this tutorial.

Summary

Summary

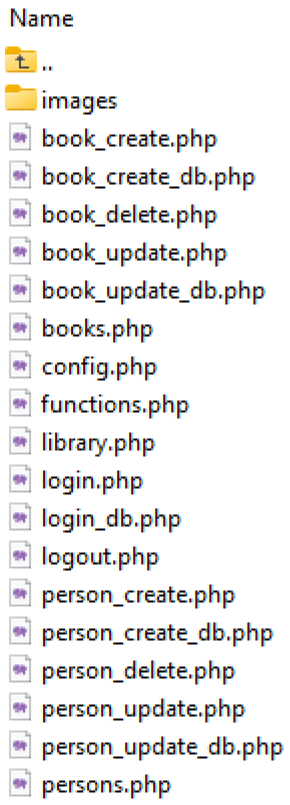


Library

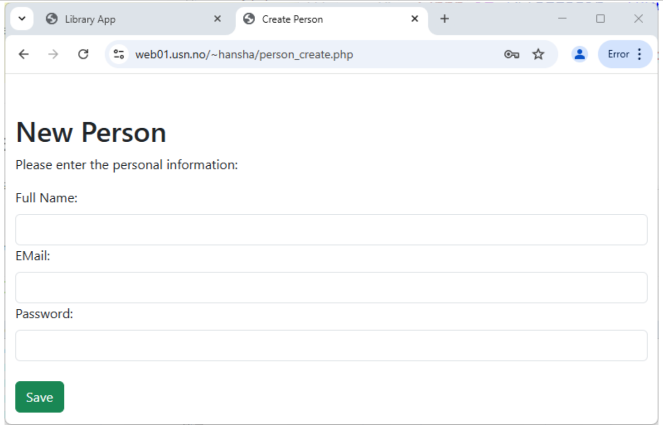
This is the Library App. Here you can see avalible books, create new books, etc.

You need to Login into the system to get access to the functionality.

We have created a basic "Library" PHP Web Application. The Application shows a list of Books, but only after the User has logged into the system. We have also created basic User Administration where we can add, delete and modify Users.



Name
..
images
book_create.php
book_create_db.php
book_delete.php
book_update.php
book_update_db.php
books.php
config.php
functions.php
library.php
login.php
login_db.php
logout.php
person_create.php
person_create_db.php
person_delete.php
person_update.php
person_update_db.php
persons.php



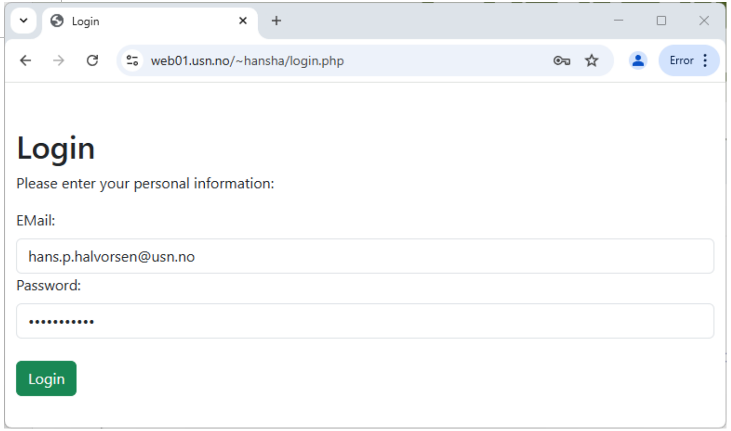
New Person

Please enter the personal information:

Full Name:

EMail:

Password:



Login

Please enter your personal information:

EMail: hans.p.halvorsen@usn.no

Password: ••••••••••••

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
- Bootstrap Tutorial: https://www.w3schools.com/bootstrap5/

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: https://www.halvorsen.blog