

INLOGGNINGSSYSTEM VIA DROPDOWN MENY

STEG 1

Följande mappar och filer ska finnas för inloggningssystemet:

- assets/
 - o config/
 - db.php
 - o css/
 - all.min.css
 - bootstrap.min.css
 - login.css
 - o functions/
 - session.check.php
 - session.login.php
 - o images/
 - o includes/
 - notifications.index.php
 - o webfonts/
- index.php
- logout.php
- protected.php

STEG 2

- Installera Bootstrap
 - o npm install bootstrap
 - o Flytta **bootstrap.min.css** till **assets/css/**
- Installera Font-Awesome
 1. npm i --save @fortawesome/fontawesome-free
 2. Öppna mappen **node_modules/@fortawesome/fontawesome-free**
 3. Flytta mappen **webfonts/** till **assets/**.
 4. Flytta filen **css/all.min.css** till **assets/css/**.
 5. Ta bort mappen **node_modules/**
 - o Ikoner finns här: <https://fontawesome.com/icons?d=gallery&m=free>
- Skapa filen **login.css** i **assets/css/**

STEG 3

- Skapa ett grundläggande HTML dokument
- Länka alla CSS filer med hjälp av <link> elementet
- Lagg in JavaScript kod innan </body> för att få dropdown funktionalitet

```
<!DOCTYPE html>
<html lang="sv">
<head>
  <meta charset="utf-8">
  <title>Login system</title>
  <!-- Bootstrap Core CSS -->
  <link rel="stylesheet" href="assets/css/bootstrap.min.css">
  <!-- Font-Awesome Core CSS -->
  <link rel="stylesheet" href="assets/css/all.min.css">
  <!-- Login CSS -->
  <link rel="stylesheet" href="assets/css/login.css">
</head>
<body>

  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.min.js"></script>
</body>
</html>
```

STEG 4

4.1. Skapa en navigation (navbar) i ett <header> element

```
<nav class="navbar navbar-expand-md navbar-dark bg-dark">
  <div class="container">
    <a href="index.php" class="navbar-brand">
      <i class="fas fa-key"></i> Login system
    </a>
  </div>
```

** HÄR.**

```
</nav>
```

4.2. Lägg till länkar och en dropdown till höger med

```
<ul class="navbar-nav">
  <li class="nav-item">
    <a href="#" class="nav-link">Hem</a>
  </li>
  <li class="nav-item">
    <a href="#" class="nav-link dropdown-toggle" id="login" data-toggle="dropdown">
      <i class="fas fa-user-circle"></i> Logga in
    </a>
```

FORMULÄR HÄR. SE PUNKT 4.3

```
</li>
</ul>
```

4.3. Skapa ett formulär för e-post och lösenord

```
<form action="index.php" method="post" class="dropdown-menu dropdown-menu-right
p-4">
  <div class="form-group">
    <label for="email" class="sr-only">E-postadress</label>
    <input type="email" class="form-control" name="email" placeholder="E-postadress">
  </div>
  <div class="form-group">
    <label for="password" class="sr-only">Lösenord</label>
    <input type="password" class="form-control" name="password"
placeholder="Lösenord">
  </div>
  <button type="submit" class="btn btn-primary btn-block" name="login">Logga
in</button>
</form>
```

4.4. Öppna assets/css/login.css och justera bredden för dropdownmenyn

```
.dropdown-menu {
  width: 300px !important;
}
```

STEG 5

Lägg till sessioner, databasuppkoppling och kontroll av inloggning överst på index.php

```
<?php
    // Init session management
    session_start();
    // Opens database connection
    require_once 'assets/config/db.php';
    // Process form data to database
    require_once 'assets/functions/session.login.php';
?>
```

STEG 6

Öppna filen assets/config/db.php och ange kod för att öppna databasuppkoppling

```
<?php
    // Defines information for database connection
    define('DB_HOST', 'localhost');
    define('DB_NAME', 'codeanywhere');
    define('DB_USER', 'root');
    define('DB_PASS', 'root');

    // Connects to database
    $dbh = new PDO(
        'mysql:host='.DB_HOST.';dbname='.DB_NAME.';charset=utf8',
        DB_USER,
        DB_PASS,
        array(
            PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
            PDO::ATTR_EMULATE_PREPARES => false,
            PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
            PDO::MYSQL_ATTR_FOUND_ROWS => true
        )
    );
?>
```

STEG 7

Gå till **phpMyAdmin** och skapa en ny tabell som kallas **login**.

Den består av fyra fält:

- **id** (int, 10) UNSIGNED, PRIMARY, AI
- **email** (varchar, 255)
- **password** (varchar, 255)
- **regdate** (datetime)

Lägg till ett konto (användarnamn: **test@hkr.se** / lösenord: **test**) via INSERT tabben.

STEG 8

Öppna assets/functions/session.login.php och ange funktion för inloggning

```
<?php
// Checks whether submit button has been set
if (isset($_POST['login'])) {
    // Checks whether e-mail or password are empty
    if (empty($_POST['email']) || empty($_POST['password'])) {
        // Redirect user to error page
        header('Location: ../../index.php?login=empty');
        exit();
    }

    // Trims e-mail and password
    $email = strtolower(trim($_POST['email']));
    $password = trim($_POST['password']);

    // Creates, prepares, binds and executes a query
    $sql = 'SELECT * FROM login WHERE email = :email AND password = :password';
    $stmt = $dbh->prepare($sql);
    $stmt->bindValue(':email', $email);
    $stmt->bindValue(':password', $password);
    $stmt->execute();

    // Counts rows returned from database
    $count = $stmt->rowCount();

    // Checks whether user exists
    if ($count > 0) {
        // Saves results to variable
        $row = $stmt->fetch();
        // Creates session variable with user id and redirects user
        $_SESSION['id'] = $row['id'];
        // Redirects user to success page
        header('Location: ../../protected.php');
        exit();
    } else {
        // Redirect user to error page
        header('Location: ../../index.php?login=error');
        exit();
    }
}
?>
```

STEG 9

Skapa feedback på index.php

```
<?php
    // Checks if an action is set
    if (isset($_GET['login'])) {
        // Checks which action is set
        switch ($_GET['login']) {
            case 'empty':
                echo '
                    <div class="alert alert-warning">
                        Du har inte angett något användarnamn eller lösenord!
                    </div>
                ';
                break;
            case 'error':
                echo '
                    <div class="alert alert-danger">
                        Du har angett felaktigt användarnamn eller lösenord!
                    </div>
                ';
                break;
        }
    }
?>
```


STEG 10

Flytta koden för feedback från index.php till assets/includes/notifications.index.php och inkludera denna filen igen till index.php med funktionen require_once

```
<main class="container mt-5">
  <?php
    // Prints out possible notifications from login process
    require_once 'assets/includes/notifications.index.php';
  ?>
</main>
```

STEG 11

Öppna logout.php och ange funktion för utloggning

```
<?php
  // Logs out user from system
  session_start();
  session_destroy();
  header('Location: index.php');
  exit();
?>
```

STEG 12

Öppna `assets/functions/session.check.php` och ange funktion som kontrollerar om en session existerar

```
<?php
    // Checks whether session id exists
    if (!isset($_SESSION['id']) || empty($_SESSION['id'])) {
        // Redirect user to front / login page
        header('Location: ../../index.php?login=notexist');
        exit();
    }
?>
```

STEG 13

Uppdatera filen `assets/includes/notifications.index.php` med feedback om att man måste logga in för att komma åt ett innehåll

```
case 'notexist':
    echo '
        <div class="alert alert-primary">
            Du måste logga in för att komma åt innehållet!
        </div>
    ';
    break;
```

STEG 14

Kopiera allt från index.php till protected.php och ange följande kod överst på protected.php för att skydda den.

```
<?php
    // Init session management
    session_start();
    // Opens database connection
    require_once 'assets/config/db.php';
    // Enables restricted access
    require_once 'assets/functions/session.check.php';
?>
```

STEG 15

Justera navigationen för att göra det möjligt för användaren att logga ut

```
<nav class="navbar navbar-expand-md navbar-dark bg-dark">
    <div class="container">
        <a href="index.php" class="navbar-brand">
            <i class="fas fa-key"></i> Login system
        </a>
        <ul class="navbar-nav">
            <li class="nav-item">
                <a href="protected.php" class="nav-link">Hem</a>
            </li>
            <li class="nav-item">
                <a href="logout.php" class="nav-link">
                    <i class="fas fa-sign-out-alt"></i> Logga ut
                </a>
            </li>
        </ul>
    </div>
</nav>
```

STEG 16

Ta bort allt innehåll i <main> inuti protected.php och ange följande text:

```
<main class="container mt-5">
    Välkommen till den skyddade sidan!
</main>
```

STEG 17

Öppna filen `logout.php` igen och lägga till en GET-variabel som gör det möjligt att skriva ut feedback vid utloggning

```
<?php
    // Logs out user from system
    session_start();
    session_destroy();
    header('Location: index.php?login=logout');
    exit();
?>
```

STEG 18

Uppdatera filen `assets/includes/notifications.index.php` med feedback när man har loggat ut

```
case 'logout':
    echo '
        <div class="alert alert-success">
            Du har lyckats logga ut! :-)
        </div>
    ';
    break;
```