

# Innehåll

## Webbserverprogrammering 1

|                      |   |
|----------------------|---|
| Inledning.....       | 1 |
| Syfte.....           | 2 |
| Händelseförlopp..... | 3 |
| Slutprodukt.....     | 4 |
| Övrigt.....          | 5 |

## Webbserverprogrammering 2

|                                   |    |
|-----------------------------------|----|
| Planering & GANNTT-Schema.....    | 6  |
| Uppdaterad Layout.....            | 7  |
| Admin & PHP texter.....           | 8  |
| Komplett Admin & Medlem sida..... | 9  |
| AJAX & PHP/JSON.....              | 10 |
| Säkerhet & Optimering.....        | 11 |
| Nytt GANTT-Schema.....            | 12 |
| Sammanställning.....              | 13 |
| Innehållsförteckning.....         | 14 |

Kaptiel som ska skrivas till den 1/1/22 är, Revidering/information om nytt GANTT-schema och hur det ska uppdateras t.ex genom revidering varje fredag. Samt bild på nya GANTT-Schema mallen i google sheets.

Utvärdings text och sammanfattning samt vad jag tyckte om mitt arbete och hur det kunnat förbättras.

## 1. Inledning

Här är tanken att vi ska få fram en webbsida som kan jobba imot en databas tillsammans med språken PHP och SQL som är språk som används när man ska både hämta, lägga till och ändra information som är lagrad i databasen. SQL - Standards query language är till för att skicka förfrågan till databasen. PHP är en script baserat språk som man kan göra olika finesser med.

SQL kan man skriva både direkt i databasen som i Myphpadmin, som är ett "förfixat" interface för att kunna hantera sql databaser. Man kan även skriva SQL i en php/html fil och med hjälp av php kan man sedan länka detta till databasen så att sql kommandona går till den specificerade databasen.

Hemsidan som är skapad är en grundlig och gör den effekt som den ska utföra, den kan hämta information, lagra information och allt kan ske av en annan användare.

## 2. Syfte

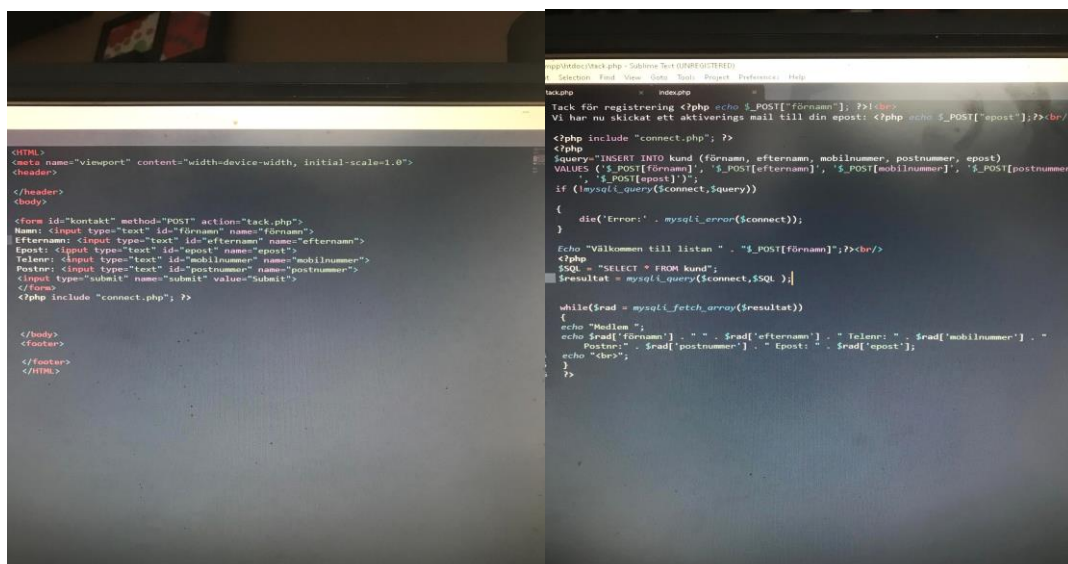
Ända målet för webbsidan är att jag själv ska få grundlig kunskap inom området och så jag kan testa och "leka" runt.

Jag ska få kunskaper om hur PHP och SQL är uppbyggt, hur jag sätter upp min egna databas, hur jag sedan lägger till, tar bort, ändrar information på sidan och hur jag kan göra detsamma fast med hjälp av PHP kod. Det stora syftet med sidan är att det ska bli en sida för en biograf.

### 3. Händelseförlopp

Denna sidan tog ändå ett litet tag att få till. Med hjälp av hemsidor och liten förståelse och kunskaperna jag fick igenom utbildningen som lyckades jag få ihopa det.

i början så skrev jag all kod i en sida och ingenting blev rätt, fick ut viss information på samma sida och såg att något fungerade, men fick undefined array key som error och tillslut insåg jag att det var pga att den array key jag ville printa ut hade inte fått något värde och när jag skrev in värdet i ett input form så blev jag skickat till en annan sida och det som skulle printas, printades inte. Delade upp koderna och fick ut det jag ville.



```

tack.php
index.php

Tack för registrering <?php echo $_POST['förnamn']; ?><br>
Vi har nu skickat ett aktiverings mail till din epost: <?php echo $_POST['epost'];?><br>

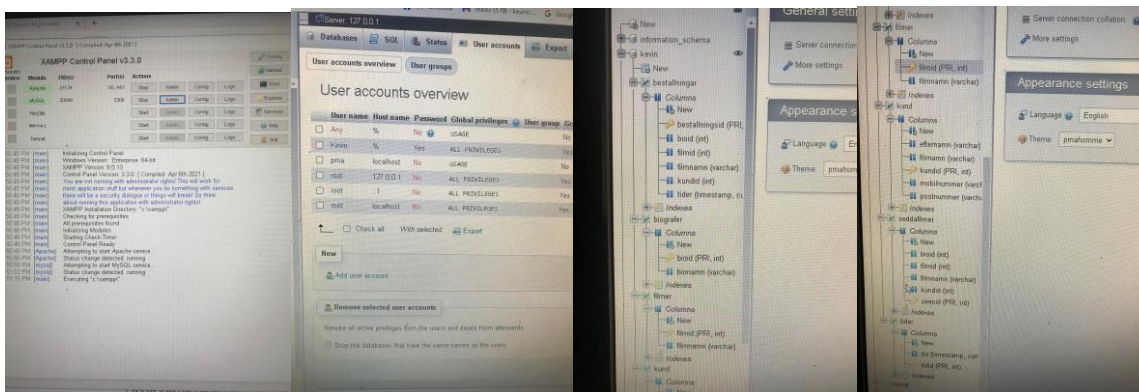
<?php include "connect.php"; ?>
<?php
$query="INSERT INTO kund (förnamn, efternamn, mobilnummer, postnummer, epost)
VALUES ('$_POST[förnamn]', '$_POST[efternamn]', '$_POST[mobilnummer]', '$_POST[postnummer]', '$_POST[epost]');";
if (!mysqli_query($connect,$query))
{
    die('Error: ' . mysqli_error($connect));
}
Echo "Välkommen till listan " . " $_POST[förnamn]";?><br>
<?php
$sql = "SELECT * FROM kund";
$resultat = mysqli_query($connect,$sql);

while($rad = mysqli_fetch_array($resultat))
{
    echo "Medlem ";
    echo $rad['förnamn'] . " " . $rad['efternamn'] . " Telnr: " . $rad['mobilnummer'] . "
    Postnr: " . $rad['postnummer'] . " Epost: " . $rad['epost'];
    echo "<br>";
}
?>

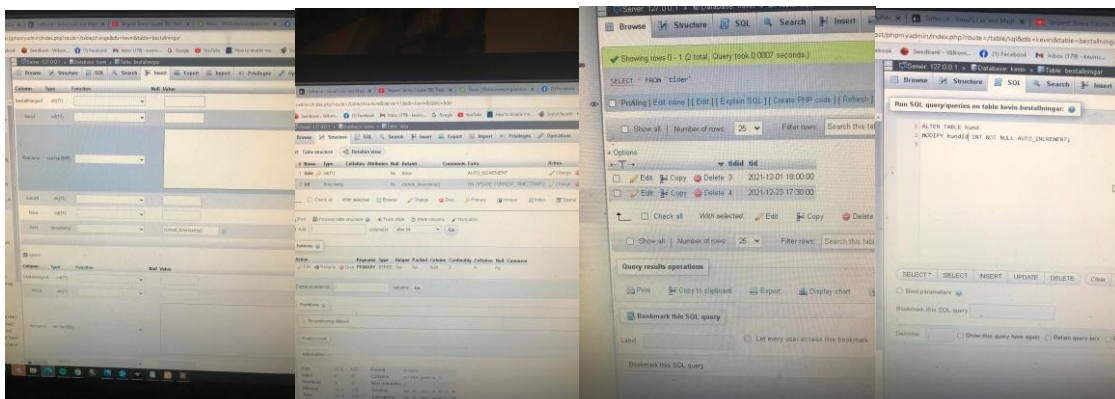
<HTML>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<header>
</header>
<body>
<form id="kontakt" method="POST" action="tack.php">
Namn: <input type="text" id="förnamn" name="förnamn">
Efternamn: <input type="text" id="efternamn" name="efternamn">
Epost: <input type="text" id="epost" name="epost">
Telnr: <input type="text" id="mobilnummer" name="mobilnummer">
Postnr: <input type="text" id="postnummer" name="postnummer">
<input type="submit" name="submit" value="Submit">
</form>
<?php include "connect.php"; ?>

</body>
</html>
  
```

3.3 Uppsättning av databasen gjordes med hjälp av XAMPP, ett färdig program för att handskas med Apache servers. Skapade konton i databasen och sedan alla tabeller gjorde i databasen som sedan ska fyllas med information som lagras och sedan kan användas i olika syften.



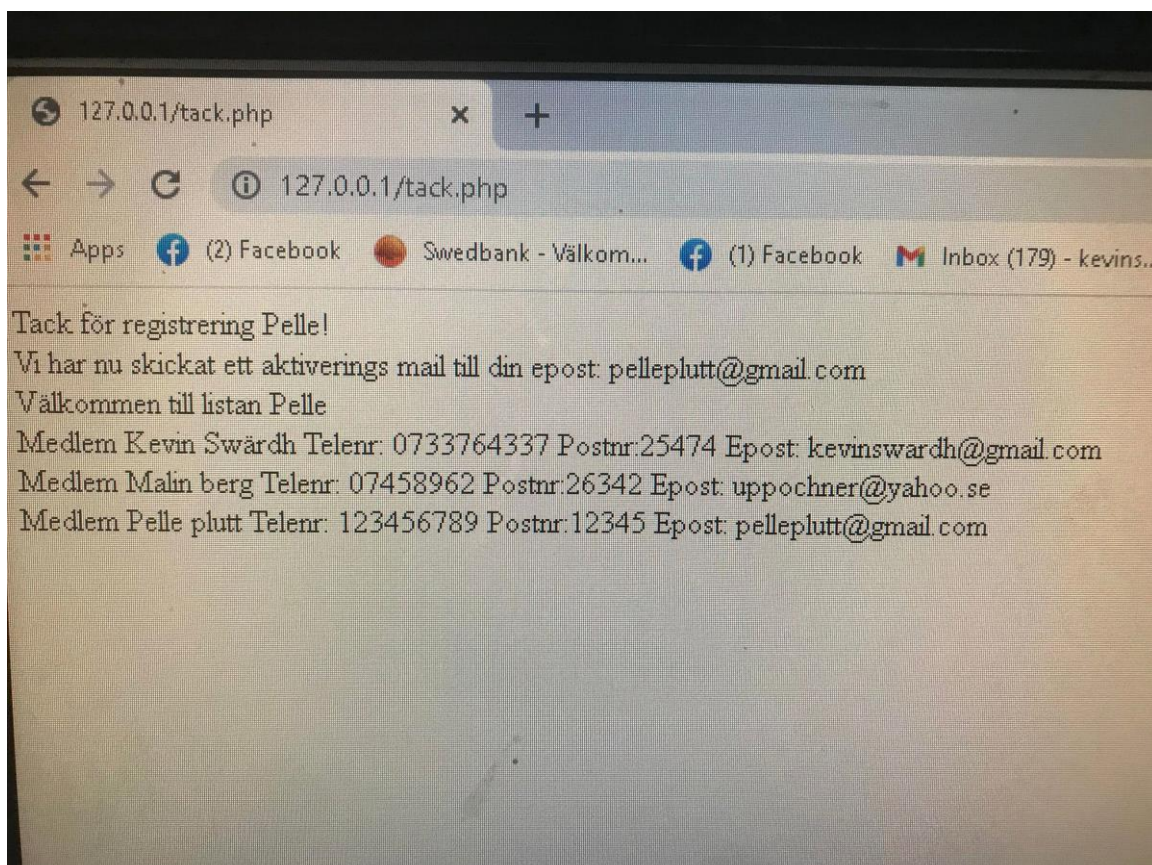
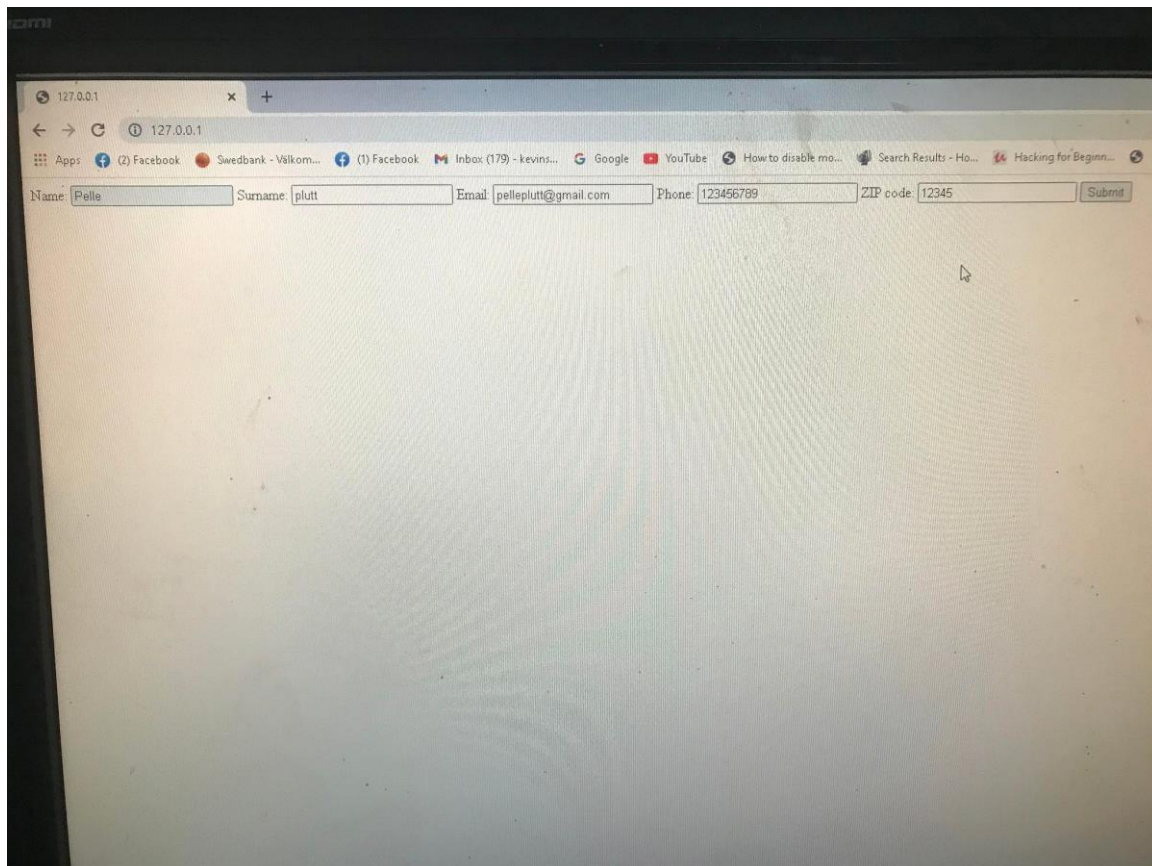
3.4 Med hjälp av olika funktioner i MyPHPAdmin som är en interface för att kunna hantera databaserna manuellt. Är där tillagt 2 kunder manuellt och tider. Detta är för att kunna testa sig runt och tiderna är för att kunna boka för inlagda tider som existerar.



## 4. Slutprodukt

Produkten blev klar tillslut och fungerar nu så att man kan skriva information, som skickas och lagras in i databasen efter man tryckt på submit så blir man skickad till en ny sida, där den sidan har genererat en ny version av html/php koden där den skriver Tack för registrering och sedan namnet som angavs, Vi har skickat en aktiveringskod till din epost och sedan eposten som angavs, undertill printas alla tidigare användare ut, som Medlem, med all information som angavs.





## 5. Övrigt

För övrigt, har jag ändå fått en del saker klargjorda, en del saker har jag fått in och en del är inte riktigt helt inne men en uppfattning om hur jag tar reda på informationen och mina vägar dit. Lärt mig hur användbart W3schools faktiskt kan vara. Fått förståelse för hur jag kan använda de olika variablerna och tycker det är jätte coolt att jag ens klara få ut den informationen som angavs i input formatet. Man vill vara bäst imorgon och helst igår, Men tyckte jag ändå fått en del grundkunskaper som går att jobba med.

## 6. Planering & GANTT-Schema

### Projekt-plan

#### PHP-Bio funktioner & Användare

##### Bakgrund

- Bakgrunden till denna hemsida valdes att göra då det kändes som en simplare uppgift i lärandet men att jag fortfarande kunde få den information och kunskap som krävdes för att kunna skapa beställningar, inlogg, användare, admin/kund sidor, och html/css sidor som varierade baserat på informationen ur databasen. Kändes som jag fortfarande kunde få den information som är grundläggande även för att göra en online butik eller chatt sidor med mera.

##### Syfte

- Syftet är att lära sig att få jobba med sessioner, säkerhet till användare, lära sig jobba med användare sidor, och admin sidor och få en större överblick över just PHP och dess funktioner.

##### Projektmål

- Målet är att få en hemsida där jag kan jobba med den dynamiskt, Så jag kan göra att mina registrerade användare nu kan logga in med den information den angett och sedan få unik information på de sidor som ger det. Målet är även att kunna göra så båda admin/användare kan ändra viss information via deras egna sidor, så som adress, namn, med mera.



## Produkt Beskrivning

- En sida för att kunna lagra, hantera och ändra information, Sessioner för inloggning och admin/kund sidor för att se sin information samt att kunna ändra den. Eventuellt göra beställning som lägg in i databasen.
- 

## Krav Specifikation

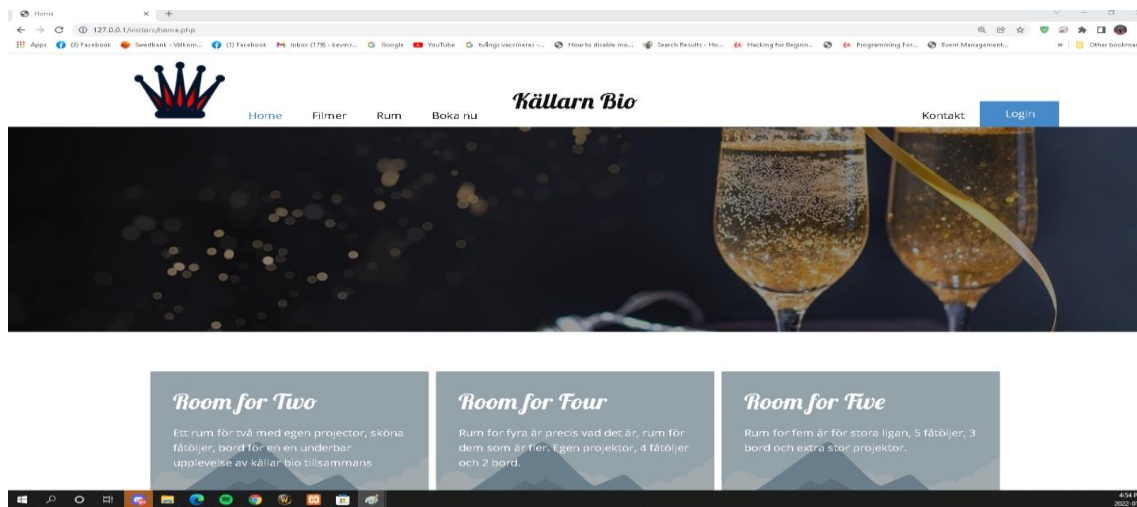
- Användare
- Sessioner
- Admin/kund sida och möjligheten att ändra information
- AJAX med JS
- JSON med jQuery
- Json och each()
- Json och externa APier
- AJAX med PHP
  - o Optimering av databas
- Apache och säkerhet
- PHP och säkerhet

## (GANTT-Schema)

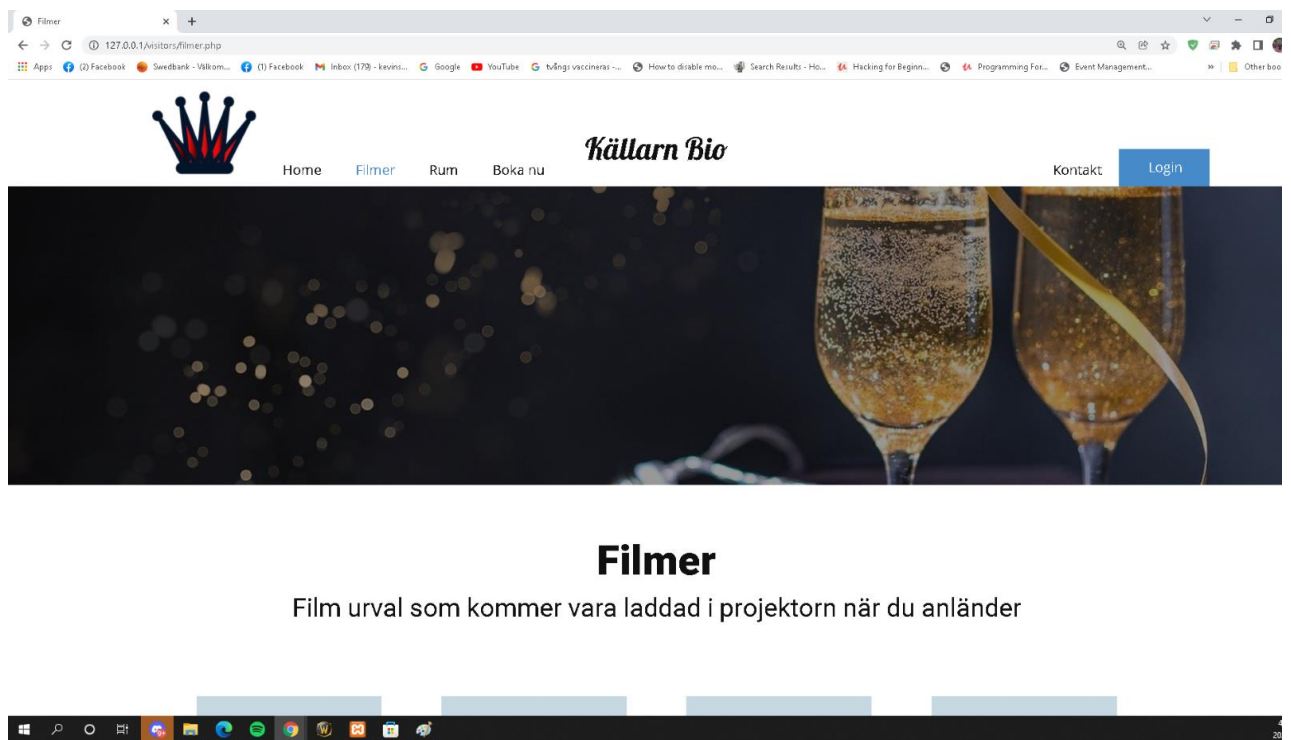


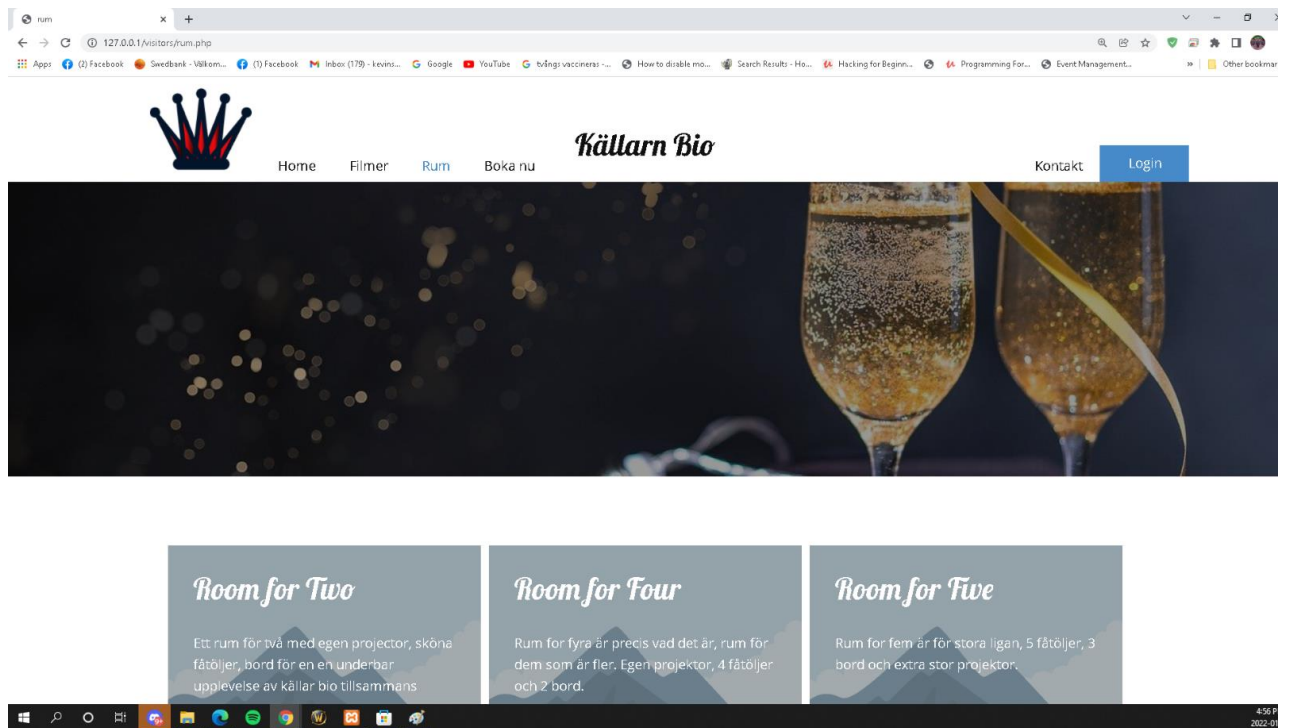
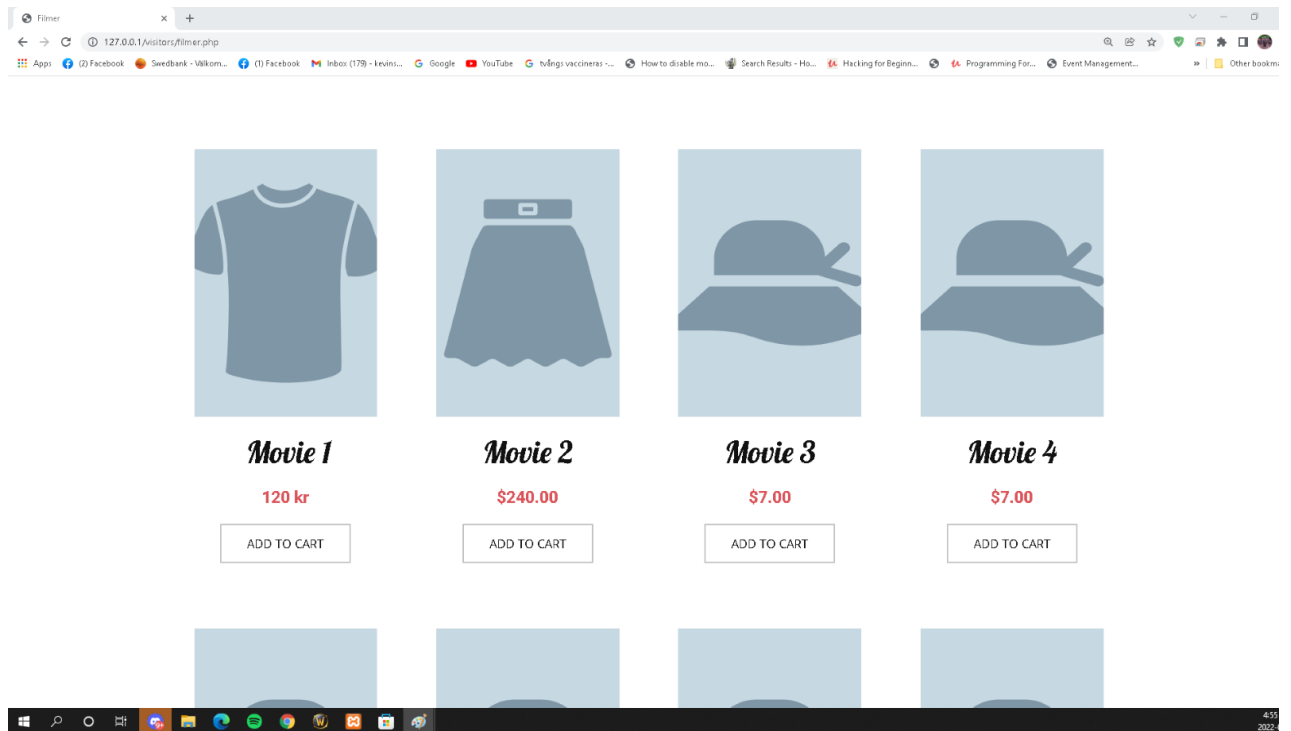
- Projektplan/start
- 15-21 Januari - Kunskaper och information gällande, min första PHP-Sida, Variabler, Villkor och loopar, Formulär och PHP, Include, Vår databas, Hämta/Lagra data, PHP data objects och slutligen Sessioner, SQL, användare och UPDATE som ger användare möjligheten att ändra. Här ska Användaren vara skapad och kunna logga in.
- 
- 21-26 Januari - Översiktligt lära sig om vad AJAX & Json är och hur det fungerar med Javascript samt PHP
- 
- 26-1 Februari - Säkerhet och optimering i databasen, PHP och Apache.
- 
- 1-6 Februari – CMS och Wordpress

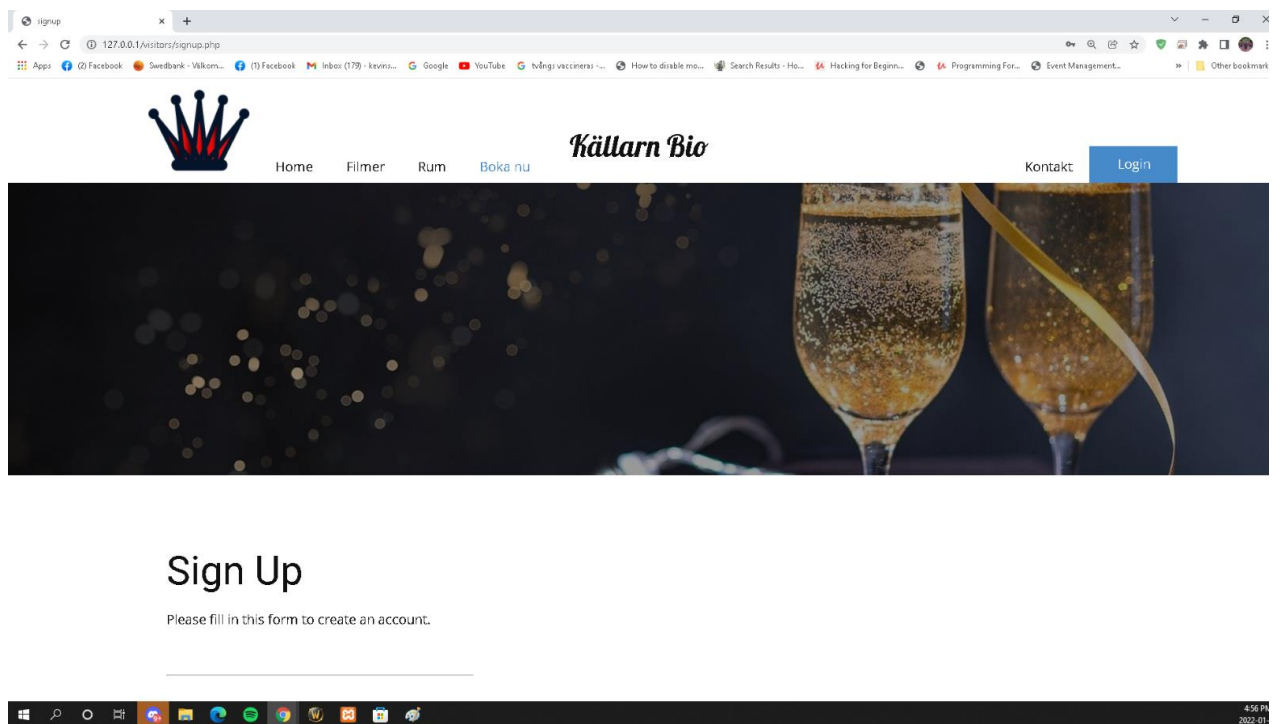
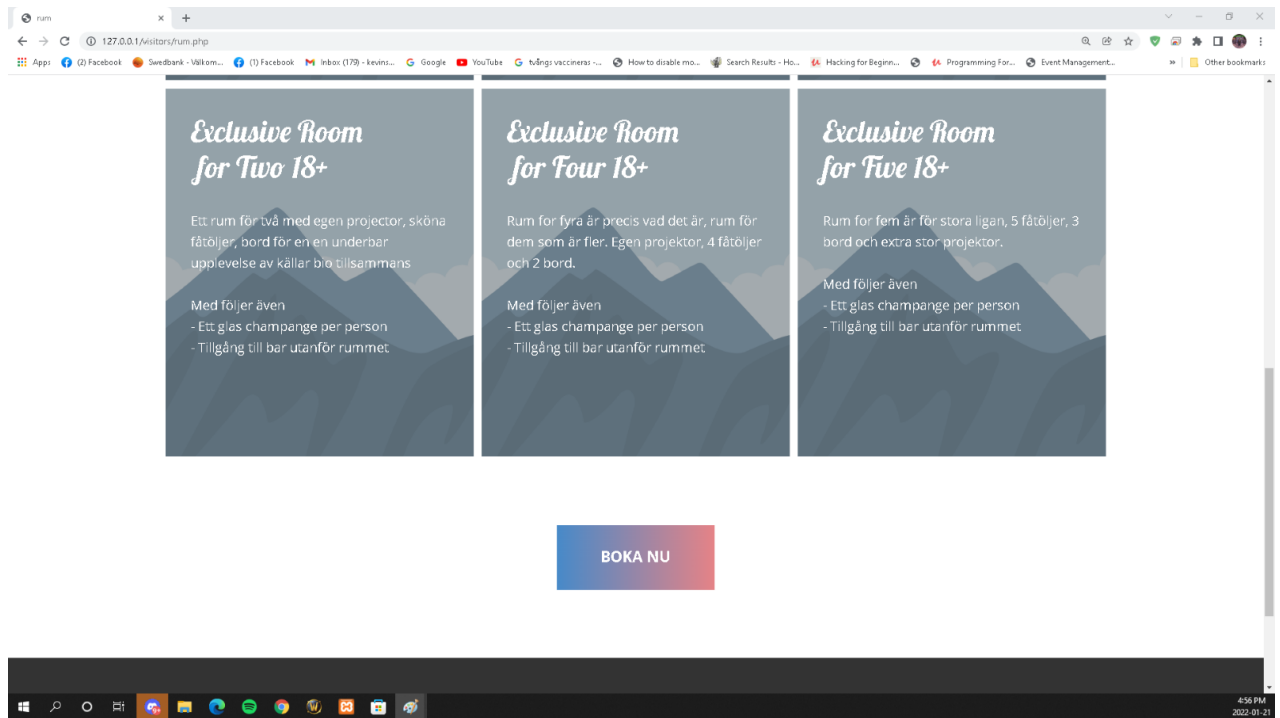
## 7. Uppdaterad Layout

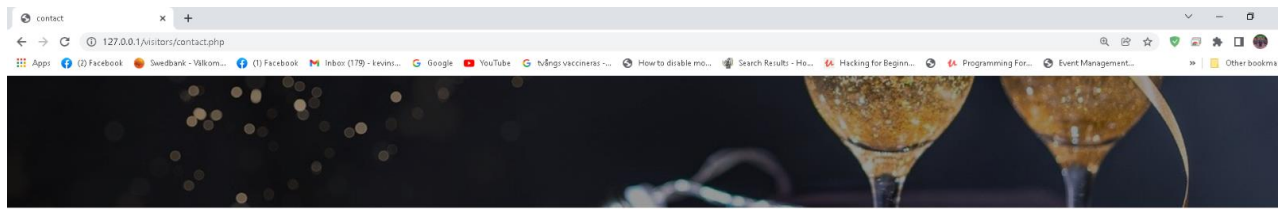


Börjat mitt skrivande med att jag fått gjort en layout till min biograf, jobbade ett litet tag med den men inte helt klar, små detaljer, hit och dit. Gjorde layouten i ett program som heter Nicepage.









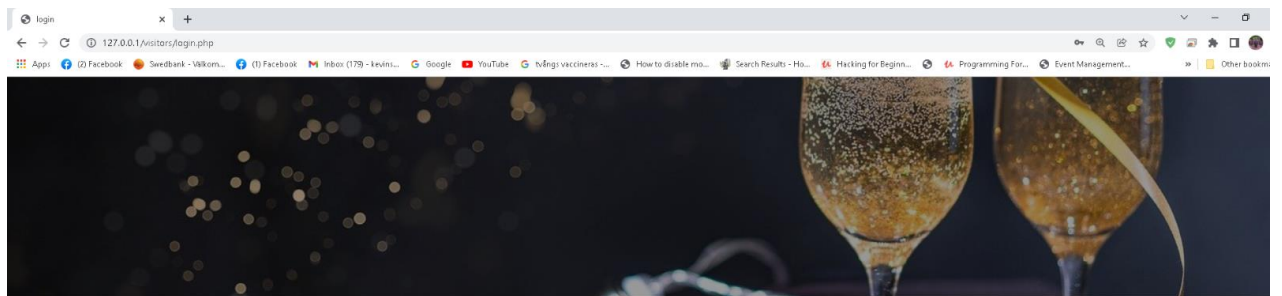
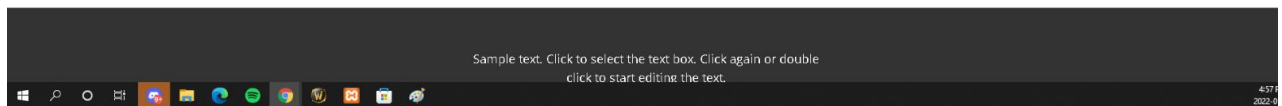
## Contact Us

Enter your Name

Enter a valid email address

Enter your message

Submit



## Log in

Please fill in this form to create an account.

kevinswa

\*\*\*\*\*

Cancel Log in

Inte medlem? [Registera dig här.](#)

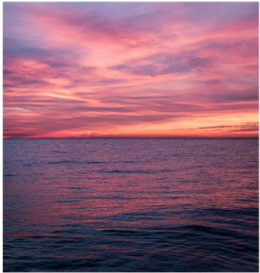




Admin sida

PROFIL ALLA BESTÄLLNINGAR

Min Profil



ÄNDRA

ÄNDRA

ÄNDRA

ÄNDRA

ÄNDRA

ÄNDRA

ÄNDRA

LOGGA UT

- Anändar namn: Kevinswa
- Lösenord:
- Namn: Kevin
- Efternamn: Swärdh
- Email: kevinswardh@gmail.com
- Stad: Helsingborg
- Telefonnummer: 073-3764337
- Postnummer: 25474

Beställning


Var god och välj film och rum nedan.

Film  
Item 1

Rum  
Item 1

Önskemål

Submit



## 8. Admin oklar & PHP Texter

### PHP

Fick PHPn att fungera tillslut med sessions() först trodde jag det fungerade men gjorde inte, sen fick jag det bara att fungera på 1 sidan man kom till efter men när man gick vidare som dog det trots jag hade session\_start på varje sida. Efter många om och men fick jag det att fungera och har en fungerande session som fungerar på alla sidor, som admin. Kunde enbart få det att fungera med mitt användarnamn/lösenord med sessionstart och håller på att titta på en kod som ser ut såhär:

```

• <?php
• /*
• session_start();
• // Change this to your connection info.
• $DATABASE_HOST = 'localhost';
• $DATABASE_USER = 'Kevin';
• $DATABASE_PASS = '9023177kevin';
• $DATABASE_NAME = 'biograf';
• // Try and connect using the info above.
• $con = mysqli_connect($DATABASE_HOST, $DATABASE_USER, $DATABASE_PASS, $DATABASE_NAME);
• if ( mysqli_connect_errno() ) {
•     // If there is an error with the connection, stop the script and display the error.
•     exit('Failed to connect to MySQL: ' . mysqli_connect_error());
• }
• // Now we check if the data from the login form was submitted, isset() will check if the data exists.
• if ( !isset($_POST['användarnamn'], $_POST['lösenord']) ) {
•     // Could not get the data that should have been sent.
•     exit('Please fill both the username and password fields!');
• }
• // Prepare our SQL, preparing the SQL statement will prevent SQL injection.
• if ($stmt = $con->prepare('SELECT användarnamn, lösenord FROM biograf WHERE användare = ?')) {
•     // Bind parameters (s = string, i = int, b = blob, etc), in our case the username is a string so we use "s"
•     $stmt->bind_param('s', $_POST['användarnamn']);
•     $stmt->execute();
•     // Store the result so we can check if the account exists in the database.
•     $stmt->store_result();
•
•
•
•     $stmt->close();
•
•
• }
•
• if ($stmt->num_rows > 0) {
•     $stmt->bind_result($användarnamn, $lösenord);
•     $stmt->fetch();
•     // Account exists, now we verify the password.
•     // Note: remember to use password_hash in your registration file to store the hashed passwords.
•     if (password_verify($_POST['användarnamn'], $lösenord)) {
•         // Verification success! User has logged-in!
•         // Create sessions, so we know the user is logged in, they basically act like cookies but remember the data on
the server.
•         session_regenerate_id();
•         $_SESSION['loggedin'] = TRUE;

```

- `$_SESSION['name'] = $_POST['användarnamn'];`
- `$_SESSION['användarnamn'] = $id;`
- `echo 'Welcome ' . $_SESSION['name'] . '!';`
- `} else {`
- `// Incorrect password`
- `echo 'Incorrect username and/or password!';`
- `}`
- `} else {`
- `// Incorrect username`
- `echo 'Incorrect username and/or password!';`
- `}`
- `*/`
- `?>`

**Men får inte det att fungera.**

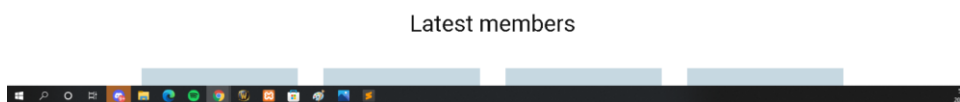
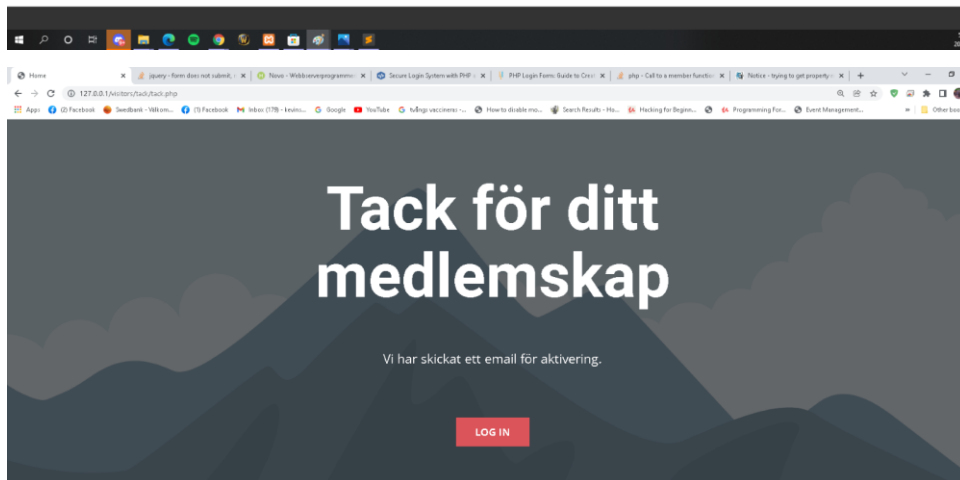
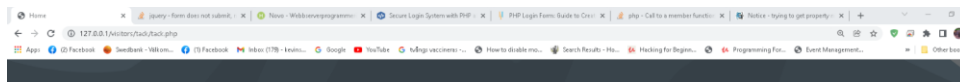
Har i alla fall gjort så här att jag har min index tsm med css i main folder, sen har jag en visitors folder, admin folder och member folder.

Page länkar i index.php går vidare till visitors foldern, i den mappen har jag alla sidor + connect.php, loghinauth.php. Login form, samt att klickar man på boka nu, kommer man till sign up. På sigup sidan blir man vidare länkad till en landingpage.php efter att man fyllt i uppgifterna

**På den står det följande:**

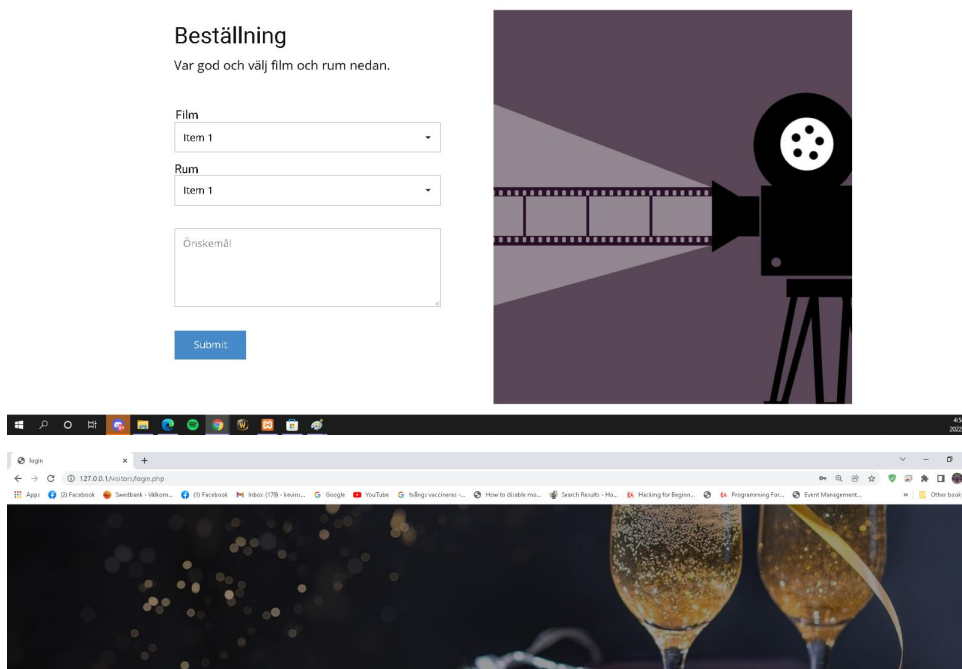
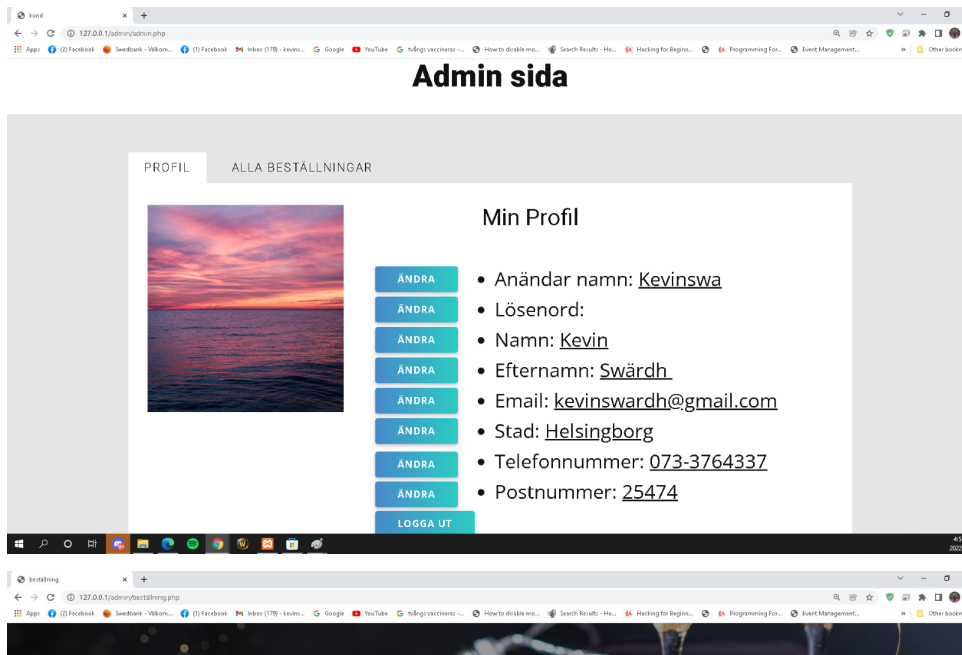
- `<?php include "../connect.php"; ?>`
- `<?php`
- `$username = $connect -> real_escape_string($_POST['användarnamn']);`
- `$password = $connect -> real_escape_string($_POST['lösenord']);`
- `$firstname = $connect -> real_escape_string($_POST['förnamn']);`
- `$secondname = $connect -> real_escape_string($_POST['efternamn']);`
- `$mobilnr = $connect -> real_escape_string($_POST['telenummer']);`
- `$postnr = $connect -> real_escape_string($_POST['postnummer']);`
- `$epost = $connect -> real_escape_string($_POST['email']);`
- `$date = "INSERT INTO användare VALUES CURRENT_TIMESTAMP";`
- `$query = "INSERT INTO användare (användarnamn, lösenord, förnamn, efternamn, telenummer, postnummer, email)`
- `VALUES ('$username', '$password', '$firstname', '$secondname', '$mobilnr', '$postnr', '$epost')";`
- `if (!mysqli_query($connect, $query))`
- `{`
- `die('Error: ' . mysqli_error($connect));`
- `}`
- `header("location: ../tack/tack.php");`
- `?>`

Den skickar sedan vidare mig till Tack sidan, som har en fin loga, en button och funktionen att den skriver ut 4 senaste medlemmar.



Efter det kan man logga in ( med det användarnamn/lösen jag angivet i loginauth.php

Där efter kan man vistas på alla sidor och istället för login står där nu Admin och ger möjligheten till admin sidan samt Boka nu ger möjligheten till beställningar.



## Log in



**PHP texter:**

- <?php
  - if (\$\_POST["användarnamn"] == "kevinswa" && \$\_POST["lösenord"] == "xxxxx")  
  { session\_start(); \$\_SESSION["Login"] = "YES";
  - \$\_SESSION["uid"] = \$\_POST["användarnamn"];  
  header("Location: ../admin/admin.php");
  - }
  - else
  - { }
  - ?>
- 

Det jag använde mig speciellt här var att jag använde mig av DESC och LIMIT 1, och sedan OFFSET med en ökning så jag fick olika resultat.

- <section class="u-align-center u-clearfix u-section-2" id="sec-2c85">
- <div class="u-clearfix u-sheet u-sheet-1">
- <h2 class="u-text u-text-default u-text-1">Latest members</h2>
- <div class="u-expanded-width u-list u-list-1">
- <div class="u-repeater u-repeater-1">
- <div class="u-align-center u-container-style u-list-item u-repeater-item">
- <div class="u-container-layout u-similar-container u-valign-top u-container-layout-1">
- 
- <h5 class="u-text u-text-2"><?php \$SQL = "SELECT \* FROM användare ORDER BY datum DESC LIMIT 1;"; /\*PHP Section for name print\*/
- \$resultat = mysqli\_query(\$connect,\$SQL);
- while(\$rad = mysqli\_fetch\_array(\$resultat))
- {
- echo \$rad['användarnamn'];
- echo "<br>";
- }
- ?></h5>
- <p class="u-text u-text-3">New Member</p>
- </div>
- </div>
- <div class="u-align-center u-container-style u-list-item u-repeater-item">
- <div class="u-container-layout u-similar-container u-valign-top u-container-layout-2">



- ``
- `<h5 class="u-text u-text-4"><?php $SQL = "SELECT * FROM användare ORDER BY datum DESC LIMIT 1 OFFSET 1;";;`
- `$resultat = mysqli_query($connect,$SQL );`
- `while($rad = mysqli_fetch_array($resultat))`
- `{`
- `echo $rad['användarnamn'];`
- `echo "<br>";`
- `}`
- `?></h5>`
- `<p class="u-text u-text-5">New Member</p>`
- `</div>`
- `</div>`
- `<div class="u-align-center u-container-style u-list-item u-repeater-item">`
- `<div class="u-container-layout u-similar-container u-valign-top u-container-layout-3">`
- ``
- `<h5 class="u-text u-text-6"><?php $SQL = "SELECT * FROM användare ORDER BY datum DESC LIMIT 1 OFFSET 2;";;`
- `$resultat = mysqli_query($connect,$SQL );`
- `while($rad = mysqli_fetch_array($resultat))`
- `{`
- `echo $rad['användarnamn'];`
- `echo "<br>";`
- `}`
- `?></h5>`
- `<p class="u-text u-text-7">New Member</p>`
- `</div>`
- `</div>`
- `<div class="u-align-center u-container-style u-list-item u-repeater-item">`
- `<div class="u-container-layout u-similar-container u-valign-top u-container-layout-4">`
- ``
- `<h5 class="u-text u-text-8"><?php $SQL = "SELECT * FROM användare ORDER BY datum DESC LIMIT 1 OFFSET 3;";;`
- `$resultat = mysqli_query($connect,$SQL );`
- `while($rad = mysqli_fetch_array($resultat))`
- `{`
- `echo $rad['användarnamn'];`

- `echo "<br>";`
- `}`
- `?></h5>`
- `<p class="u-text u-text-9">New Member</p>`
- `</div>`
- `</div>`
- `</div>`
- `</div>`
- `</div>`
- `</section>`

#### Detta använda jag i början på alla sidor

- `<?php include "../connect.php"; ?>`
- `<?php`
- `session_start();`
- 
- 
- `if (isset($_SESSION))`
- `{`
- `}`
- `else`
- `{`
- `header("location: ../visitor/login.php");`
- `}`
- `?>`

Kände att det blev mestadels språktext och bilder i detta dokument, men suttit som en däre innan allt fungerade, och har nog snart inget hår på huvudet 😊

Men i grund och botten gjorde jag:

- Ny layout
- Använde mig av gamla idén att ha lista efter form input. Med en mellansida som kallas landingpage.php och sedan en tack sida med SQL förfrågan som jag sedan utvecklade för att få det resultat jag specifikt vill ha ist för radar lista uppifrån och ned så fick jag nu tsm med datum som genereras vid submit och order by samt DESC, LIMIT och OFFSET det resultat jag sökte.
- Gjorde en successful session:start() från ett annat input form. Där sessionen fortsätter på varje sida. Tills du klickar logout för då gör den onclick sess\_destroy + länkas till Visitors/home.php
- När användaren är inloggad har den då tillgång till fler pages och knappen login ändras.

Saker jag än ska lägga till är

- Sesson\_start för något som passar med databasen (members)
- Password\_hash eller liknande funktion.
- Beställnings tabellerna på flikerna beställningar
- Möjligheten att ändra sitt namn på sin sida.

## 9. Kompletta Admin & Medlem sida

Fått min admin/kund sida att fungera. Där jag antingen kan logga in som admin eller som en registrerad användare. Får då tillgång till sidor som antingen admin page eller member page beroende på vilket konto man är inne på och man får tillgång till beställnings sidan oavsett vilken man är på.

Folder uppbyggnad ser ut så här.

Main.Folder-----Visitors.Folder-----Members.Folder-----Admin.Folder

**Main folder innehåller:** Welcome page

**Visitors Foldern innehåller:** Home page, Filmer, Rum, Signup(bokanu), Kontakt, Login. Loginauth, landingpage(signupauth), Tack page.

**Members Folder innehåller:** Samma uppsättning men Signup(bokanu) ersätter beställningmember och login ersätter member page och utan login/signup.

**Admin Folder innehåller:** Samma uppsättning men Signup(bokanu) ersätter beställningmember och login ersätter admin page och utan login/signup.

### 9.1 Inloggningsystem

**Kod uppbyggnad för inloggningsystemet ser ut så här:**

**Signup.php**

- **<!--Form -->**
- `<div class="all">`
- `<form id="kontakt" method="POST" action="landingpage.php" style="max-width: 370px""border:1px solid #ccc">`
- `<div class="container">`

- <h1>Sign Up</h1>
- <p>Please fill in this form to create an account.</p></br>
- <hr>
- 
- <label for="username"></label></br>
- <input type="text" placeholder="Ange användar namn" id="användarnamn" name="användarnamn" required>
- 
- <label for="psw"></label></br>
- <input type="password" placeholder="Ange lösenord" id="lösenord" name="lösenord" required>
- 
- <label for="psw-repeat"></label></br>
- <input type="password" placeholder="Upprepa lösenord" id="lösen-repeat" name="lösen-repeat" required>
- 
- <label for="epost"></label></br>
- <input type="text" placeholder="Ange epost" id="email" name="email" required>
- 
- <label for="namn"></label></br>
- <input type="text" placeholder="Ange förnamn" id="förnamn" name="förnamn" required>
- 
- <label for="eternamn"></label></br>
- <input type="text" placeholder="Ange efternamn" id="eternamn" name="eternamn" required>
- 
- <label for="telenr"></label></br>
- <input type="number" placeholder="Ange telefon nummer" id="telenummer" name="telenummer" required>
- 
- <label for="postnr"></label></br>
- <input type="number" pattern="[0-9]{5}" placeholder="Ange post nummer" id="postnummer" name="postnummer" required>
- 
- 
- 
- <label></br>
- <input type="checkbox" checked="checked" name="remember" style="margin-bottom:15px"> Remember me
- </label></br>
- 
- <p>By creating an account you agree to our <a href="#" style="color:dodgerblue">Terms & Privacy</a>.</p>
- 
- <div class="clearfix">
- <button type="button" class="cancelbtn" onclick="location.href='login.php'" >Log in</button>
- <button type="submit" class="signupbtn">Sign Up</button> </br>
- </div>
- 
- </form>

## Landingpage.pgp (signupauth)

- <?php include "../connect.php"; ?>
- 
- <?php
- 
- 
- \$username = \$connect -> real\_escape\_string(\$\_POST['användarnamn']);
- \$password = \$connect -> real\_escape\_string(\$\_POST['lösenord']);
- \$firstname = \$connect -> real\_escape\_string(\$\_POST['förnamn']);
- \$secondname = \$connect -> real\_escape\_string(\$\_POST['efternamn']);
- \$mobilnr = \$connect -> real\_escape\_string(\$\_POST['telenummer']);
- \$postnr = \$connect -> real\_escape\_string(\$\_POST['postnummer']);
- \$sepost = \$connect -> real\_escape\_string(\$\_POST['email']);
- 
- \$hashed\_password = password\_hash(\$password, PASSWORD\_DEFAULT);
- \$date = "INSERT INTO användare VALUES CURRENT\_TIMESTAMP";
- \$query="INSERT INTO användare (användarnamn, lösenord, förnamn, efternamn, telenummer, postnummer, email)
- VALUES ('\$username', '\$hashed\_password', '\$firstname', '\$secondname', '\$mobilnr', '\$postnr', '\$sepost')";
- 
- if (!mysqli\_query(\$connect,\$query))
- 
- {
- die('Error: ' . mysqli\_error(\$connect));
- }
- header("location: ./tack/tack.php");
- ?>

## Login.php

- <div class="all">
- <form id="kontakt" method="POST" action="loginauth.php" style="max-width: 370px""border:1px solid #ccc";>
- <div class="container">
- <h1></h1>
- <p>Please fill in this form to create an account.</p>
- <hr>
- 
- <label for="username"></label>
- <input type="text" placeholder="Ange användar namn" id="användarnamn" name="användarnamn" required>
- 
- <label for="psw"></label>



- `<input type="password" placeholder="Ange lösenord" id="lösenord" name="lösenord" required>`
- 
- `<div class="clearfix">`
- `<button type="button" class="cancelbtn">Cancel</button>`
- `<button type="submit" class="signuptbn">Log in</button>`
- `</div>`
- `</div>`
- `</form>`

## Loginauth.php

- `<?php include "../connect.php"; ?>`
- 
- `<?php //Kolla användarnamn och lösenord från ett formulär`
- `if ($_POST["användarnamn"] == "kevinswa" && $_POST["lösenord"] == "9023177kevin") { // Om sant sätter vi`  
`Login till YES session_start(); $_SESSION["Login"] = "YES";`
- `$_SESSION["uid"] = $_POST["användarnamn"]; header("Location: ../admin/admin.php");`
- `}`
- **else**
- `{`
- `session_start();`
- `// Change this connection setting to your preference.`
- `$DATABASE_HOST = 'localhost';`
- `$DATABASE_USER = 'Members';`
- `$DATABASE_PASS = 'XXXXXX';`
- `$DATABASE_NAME = 'biograf';`
- `// Try and connect using the info above.`
- `$con = mysqli_connect($DATABASE_HOST, $DATABASE_USER, $DATABASE_PASS,`  
`$DATABASE_NAME);`
- `if ( mysqli_connect_errno() ) {`
- `// If there is an error with the connection, stop the script and display the error.`
- `exit('Failed to connect to MySQL: ' . mysqli_connect_error());`
- `}`
- `// Time to check if the login form data was submitted. The isset() function checks if the form data exists.`
- `if ( !isset($_POST['användarnamn'], $_POST['lösenord']) ) {`
- `// Could not fetch any data from form submission`
- `exit('Please make sure you filled both the username and password form fields!');`
- `}`
- `// We need to Prepare our SQL. This SQL preparation helps prevent SQL injections`
- `if ($stmt = $con->prepare("SELECT användarid, lösenord FROM användare WHERE användarnamn = ?")) {`
- `// Bind parameters (s = string, i = int, b = blob, etc). Since a string is the username in our case, we use "s"`
- `$stmt->bind_param('s', $_POST['användarnamn']);`
- `$stmt->execute();`
- `// Store or preserve the results. It helps counter-check if the user account exists within our database.`

```

• $stmt->store_result();
•
•
• }
• if ($stmt->num_rows > 0) {
•   $stmt->bind_result($id, $password);
•   $stmt->fetch();
•   // At this point, the account exists. The only thing left is to verify the password.
•   // The use of password_hash in the registration file is encouraged for the storage of hashed passwords.
•
•   if (password_verify($_POST['lösenord'], $password)) {
•     // Verification was a success! Use log in took place!
•     // Sessions creation takes place because a user is logged in. Sessions functionality resemble cookies because they
      can remember the server's data.
•     session_regenerate_id();
•     $_SESSION['loggedin'] = TRUE;
•     $_SESSION['name'] = $_POST['användarnamn'];
•     $_SESSION['id'] = $id;
•     echo 'Welcome ' . $_SESSION['name'] . '!';
•     header('Location: ../members/member.php');
•   } else {
•     // Incorrect password
•     echo 'Incorrect username and/or password!';
•   }
•   } else {
•     // Incorrect username
•     echo 'Incorrect username and/or password!';
•   }
•   }
•   ?>

```

### Följt av för admin på alla admin sidor

```

• <?php
• // We need to use sessions, so you should always start sessions using the below code.
• session_start();
• // If the user is not logged in redirect to the login page...
• if (!isset($_SESSION['uid'])) {
•   header('Location: admin.php');
•   exit;
• }
• ?>

```

### Och för member på alla member sidor

- <?php
- // We need to use sessions, so you should always start sessions using the below code.
- session\_start();
- // If the user is not logged in redirect to the login page...
- if (!isset(\$\_SESSION['loggedin'])) {
- header('Location: filmer.php');
- exit;
- }
- ?>

## Där jag har

header('Location: admin.php');

## För respektive sida

header('Location: filmer.php');

## Admin page/Member page

Där tanken är att jag ska göra ett session namn för varje attribut email/namn/efternamn m.m

- <li>Anändar namn: <span style="text-decoration: underline !important;"><?php echo  
\$\_SESSION['name'] ?> </span>
- </li>
- <li class="lösen">Lösenord:</li>
- <li>Namn: <span style="text-decoration: underline !important;"></span>
- </li>
- <li>Efternamn: <span style="text-decoration: underline !important;">Swärdh&nbsp;</span>
- </li>
- <li>Email: <span style="text-decoration: underline !important;">kevinswardh@gmail.com</span>
- </li>
- <li>Stad: <span style="text-decoration: underline !important;">Helsingborg</span>
- </li>
- <li>Telefonnummer: <span style="text-decoration: underline !important;">073-3764337</span>
- </li>
- <li>Postnummer: <span style="text-decoration: underline !important;">25474</span>
- </li>
- </ul>

## 10. AJAX & PHP/JSON

AJAX & PHP: Gjorde en Total Pris: där det står Total Pris: Kr och sedan när man väljer vilket rum man vill sitta i hämtar den informationen från databasen och ändrar texten till då Total Pris: 123 Kr, Baserat på vilket pris som finns för respektive rad.

**Texten ser ut såhär**

### Beställningmember.php

- <script>
- function showUser(str) {
- if (str == "") {
- document.getElementById("txtHint").innerHTML = "";
- return;
- } else {
- var xmlhttp = new XMLHttpRequest();
- xmlhttp.onreadystatechange = function() {
- if (this.readyState == 4 && this.status == 200) {
- document.getElementById("txtHint").innerHTML = this.responseText;
- }
- };
- xmlhttp.open("GET","getuser.php?q="+str,true);
- xmlhttp.send();
- }
- }
- </script>

### Och även i select div

- <select id="select-c4b1" name="users" class="u-border-1 u-border-grey-30 u-input u-input-rectangle" onchange="showUser(this.value)">
- <option value="0">
- <?php echo "Välj rum";
- ?>
- </option>
- <option value="1">
- <?php \$SQL = "SELECT \* FROM rum ORDER BY rumid LIMIT 1 OFFSET 0;";
- \$resultat = mysqli\_query(\$connect,\$SQL );
- while(\$rad = mysqli\_fetch\_array(\$resultat))
- {
- echo \$rad['rumnamn'];
- echo "<br>";

```

•         }
•         ?>
•     </option>
•     <option value="2">
•         <?php $SQL = "SELECT * FROM rum ORDER BY rumid LIMIT 1 OFFSET 2;";
•         $resultat = mysqli_query($connect,$SQL );
•         while($rad = mysqli_fetch_array($resultat))
•         {
•             echo $rad['rumnamn'];
•             echo "<br>";
•         }
•         ?>
•     </option>
•     <option value="3">
•         <?php $SQL = "SELECT * FROM rum ORDER BY rumid LIMIT 1 OFFSET 3;";
•         $resultat = mysqli_query($connect,$SQL );
•         while($rad = mysqli_fetch_array($resultat))
•         {
•             echo $rad['rumnamn'];
•             echo "<br>";
•         }
•         ?>
•     </option>
•     <option value="4">
•         <?php $SQL = "SELECT * FROM rum ORDER BY rumid LIMIT 1 OFFSET 4;";
•         $resultat = mysqli_query($connect,$SQL );
•         while($rad = mysqli_fetch_array($resultat))
•         {
•             echo $rad['rumnamn'];
•             echo "<br>";
•         }
•         ?>
•     </option>
•     <option value="5">
•         <?php $SQL = "SELECT * FROM rum ORDER BY rumid LIMIT 1 OFFSET 5;";
•         $resultat = mysqli_query($connect,$SQL );
•         while($rad = mysqli_fetch_array($resultat))
•         {
•             echo $rad['rumnamn'];
•             echo "<br>";
•         }
•         ?>
•     </option>

```

- <option value="6">
- <?php \$SQL = "SELECT \* FROM rum ORDER BY rumid LIMIT 1 OFFSET 6;";
- \$resultat = mysqli\_query(\$connect,\$SQL );
- while(\$rad = mysqli\_fetch\_array(\$resultat))
- {
- echo \$rad['rumnamn'];
- echo "<br>";
- }
- ?>
- </option>
- </select>

### Där jag sedan har en php fil som heter getfilm.php

- <?php
- \$q = intval(\$\_GET['q']);
- 
- \$con = mysqli\_connect('localhost','Kevin','xxxx','biograf');
- if (!\$con) {
- die('Could not connect: ' . mysqli\_error(\$con));
- }
- 
- mysqli\_select\_db(\$con,"ajax\_demo");
- \$sql="SELECT \* FROM rum WHERE rumid = ".\$q."";
- \$result = mysqli\_query(\$con,\$sql);
- 
- echo "Total Pris: ";
- while(\$row = mysqli\_fetch\_array(\$result)) {
- 
- echo "<td>" . \$row['pris'] . "</td>";
- 
- echo "</tr>";
- }
- echo "</table>";
- echo " Kr";
- mysqli\_close(\$con);
- ?>



### Beställning


Var god och välj film och rum nedan.

Film  
Item 1

Rum  
Välj rum

Total Pris: Kr  
Önskemål

Submit




### Beställning

Var god och välj film och rum nedan.

Film  
Item 1

Rum  
Välj rum  
Välj rum  
Rum för Två  
Rum för Fem  
18+ Exclusive Rum för Två  
18+ Exclusive Rum för Fem  
18+ Exclusive Rum för Fyra

Submit



### Beställning


Var god och välj film och rum nedan.

Film  
Item 1

Rum  
Rum för Två

Total Pris: 123 Kr  
Önskemål

Submit



AJAX & JSON: Tanken är att jag också ska göra en sök funktion på filmer sidan, haft många idéer och insett att en sökfunktion inte är det optimalaste då det troligtvis på denna sidan inte lär ge en stor effekt och majoriteten av sökningarna kommer vara No result. Så tänkte mig en Kategori val möjlighet, och har än så länge bara fått det att ändra namn på de divar jag har aktivt men har jag totalt 10 filmer och där bara är 4 filmer i en kategori ändras bara 4 namn och andra divarna är kvar och ska se hur jag får dem hidden. Ut över det tycker jag JSON filer och XML är väldigt liknande men att man istället så kan lägga till olika värden i samma objekt med mera. Men sökfunktionen som jag vill göra på filmer får jag till bästa sätt via PHP också då jag kan samla filmerna i databasen där och ta informationen till flera ställen från samma. Fick även en idé att sälja t.ex API:er på nätet. Får tillägga också att `loadJSON()`, var `url = "filmer.json"`; och `JSON.prase` måste användas istället om man vill använda json filer, samt att json filer lagras såhär: `{ "name": "Sara", "age": 23, "gender": "Female", "department": "History", "car": "Honda" }` Viktigt att inte ha komma på sista. Och `[1].name`; eller `[1].gender`; eller `[2].name`; `[2].gender`; definierar vart i objektet texten ska komma ifrån.

Har förstått det som i Node.js kan man skriva egna JSON filer automatiskt vilket för mig hade gjort JSON mer funktionell för en sök funktionen, så man kan lägga till filmer efter händelse. Eller när användare skapas där det då t.ex får samma id som i databasen och på så sätt kan man lättare printa json id 34.namn och det ska stämma med t.ex inlogga id. Försöker också ta reda på om jag kan använda mig av bilder i JSON objekt. Tror också JSON är funktionell när det ska ändras information som inte behöver vara i databasen på mysql.

## 11. Säkerhet & Optimering

### 1. Apache optimering & Säkerhet

Flesta inställningar kan vara för implementerade om man använder XAMPP med mera.

Information om vart man får ta på viss information anser jag väldigt effektiv och kan ge mig möjligheter även på en linux burk,

Är väldigt intresserad av att övergå till linux, men en linux som klarar de spel jag spelar samt kommunikations program och musik program m.m men att det ändå ger möjligheten för server hantering.

#### 1. Server info

```
#LoadModule info_module modules/mod_info.so --- httpd.conf
```

- /server-info page, Server info avslöjar apache version, openssl version och då kan hackerna ta reda på om det tex är en gammal version som är sårbar för vissa exploits. Server-info ska man kunna hitta på <http://www.minhemsida.se/server-info>

#### 2. Server status

```
#<Location /server-status>
```

```
# SetHandler server-status
```

```
# Order deny,allow
```

```
# Deny from all
```

```
# Allow from .your_domain.com
```

```
#</Location>
```

- Server status, listar information om server prestration, Server uptime, server load, current http requests och client ip adresser. Sårbar information som helst inte ska vara ute för att ge hackerna en idé att börja göra något mot servern.

### 3. Server signature

ServerSignature Off

- Server signature ger information om vilket operativsystem och apache version.

### 4. ServerTokens

ServerTokens Prod

- servertokens kontrollerar information som är skickat tillbaka till serverns response header field. Och finns flertal Servertokens att använda sig av som är listade i Apaches ServerTokens Dokumentation. By default borde den dock sättas till PROD för att enbart skicka apache i response header.

### 5.

<Directory /your/website/directory>

Options -Indexes

</Directory>

- Directory listing ger dig en view över hela content och hackern kan få tag på source koden och sedan titta på hela koden efter sårbarheter.

- 6. Apache module documentantation, för att kolla om där är moduler man ej behöver från apaches förinstallerade moduler. Kan anpassas efter hemsida.

- 7. Modsecurity WAF, open-source module som fungerar som en web application firewall, modulen hjälper till med Real-time traffic monitoring också.

ModSecurity Manual för att installera. Fungerar även mot DDOS, SQL injection, Cross-site scripting tills man själv fixat flera om säkerhets bristerna på annat sätt.

## 7.1 - Modulen mod\_security & mod\_evasive

### 8. Googles pagespeed. Kollar hemsidans speed.

mod\_pagespeed

- Bildoptimering (komprimering och storleksförändring)

Ihopslagning och minimering av Javascript och CSS-kod

Cache och beständig TCP-kontakt

### 9. Modulen Mod\_cache, är behov.

### 10. Avaktivera Caching.

```
<IfModule headers_module> RequestHeader unset DNT env=bad_DNT
```

```
</IfModule>
```

```
<IfModule expires_module> ExpiresActive On ExpiresDefault "access plus 1 seconds"
ExpiresByType text/html "access plus 1 seconds"
```

```
ExpiresByType image/gif "access plus 120 minutes"
```

```
ExpiresByType image/jpeg "access plus 120 minutes"
```

```
ExpiresByType image/png "access plus 120 minutes"
```

```
ExpiresByType text/css "access plus 60 minutes"
```

```
ExpiresByType text/javascript "access plus 60 minutes"
```

```
ExpiresByType application/x-javascript "access plus 60 minutes"
```

```
ExpiresByType text/xml "access plus 60 minutes"
```

```
</IfModule>
```

## 11. Keep-Alive

KeepAliveTimeout 30

12. Komprimera filer. Minskar Content-length och ska då gå snabbare att ladda vissa sidor mer flertal bilder med mera.

```
<IfModule deflate_module>
```

```
SetOutputFilter DEFLATE
```

```
</IfModule>
```

```
#           Komprimera           inte           binära           filer           SetEnvIfNoCase
Request_URI .(?:exe|t?gz|zip|iso|tar|bz2|sit|rar) no-gzip dont-vary (filerna här är filer man t.ex
vill ha okomprimerat, som när man vill ge användaren möjlighet att ladda ner en iso fil eller
liknande.)
```

```
#Några rader från Apaches dokumentation för specifika webbläsare
```

```
BrowserMatch ^Mozilla/4 gzip-only-text/html
```

```
BrowserMatch ^Mozilla/4\.0[678] no-gzip
```

```
BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
```

13. Minify, minimerar CSS och Javascript filer.

<https://github.com/mrclay/minify>

## 2. MySQL Optimering

1. Populate mysql database with test data för att kunna göra egna tester.
2. Hittade <https://dev.mysql.com/doc/refman/8.0/en/optimization.html> för att kunna hitta mer MySQL optimering, men känns som att man lägger till småsaker efterhand att en sida blir större.
3. Hittade även indexering där jag fick läsa om Btree och hash index. och att man kan använda sig av EXPLAIN för att veta vilket index det är för en viss SELECT.
4. Minimumint(Tinyint) där det behövs. Eller smallint andra alternativ är bigint och int.
5. Var VARCHAR(255) för enbart saker som hash lösen osv och inte för person namn och användar namn, sätt minimum och max i fältet på hemsidan för att åtgärda att användaren använder sig av ett användar namn som inte är lägre eller mindre och skapar problematik att alla rader läggs in utom användarnamn och användaren kan inte logga in och inte förstår vad som händer.

### 3. Utvbloggen.

**6.1** - InnoDB är många gånger att föredra framför MyISAM. Jag fick jättemycket problem med låsningar då jag skriver och läser mycket till stora tabeller. MyISAM är generellt snabbare på INSERT's

**6.2** - Undvik typkonverteringar. I python exempelvis så görs detta automatiskt ibland och då kan det leda till att MySQL måste göra typkonverteringar vid varje jämförelse.

– "Avoid overuse of MySQL's automatic type conversion. MySQL will perform automatic type conversion, but if you can avoid conversions, you may get better performance" (från boken MySQL: The definitive guide to using, programming, and administering MySQL 4.1 and 5.0, Third Edition)

**6.3** - Använd EXPLAIN före SELECT manuellt för att se var flaskhalsar kan sitta.

**6.4** - Försök att undvika FULLTEXT indextyper, en vacker dag så sitter du där med tonvis med information och sökningarna tar flera sekunder. Snegla istället på Sphinx.

**6.5** - Tråda så mycket som möjligt. Men var inte dum.

**6.6** - Slå på log\_slow\_queries i my.cnf och kolla loggen vilka frågor som går långsamt och försök optimera dessa.



**6.7** - Bryt ner SQL-frågorna och utred vilket värde som tar lång tid att få fram.

**6.8** - Kör SHOW FULL PROCESSLIST ibland och titta på "Time" fältet där du ser hur lång tid frågor tar.

**6.9** - Undvik ORDER BY RAND() då dessa frågor kopieras till en temporär tabell vilket tar tid.

**7.0** - Ha full koll på dina INDEX så att de används på ett optimalt sätt. Inte för många och inte för få.

**7.1** - Försök att tweaka my.cnf så att den stämmer överens med just din hårdvara

gällande RAM-minne osv. Exempelvis så kan innodb\_buffer\_pool\_size vara upp till 80% av minnesstorleken.

**7.2** - Använd LIMIT där det är möjligt.

**7.3** - Testa att lägg databasen på en separat hårddisk (datadir). Är standard /var/mysql på de flesta OS.

**7.4** - Kör OPTIMIZE TABLE en gång i veckan eller en gång per månad.

**7.5** - Ställ dig frågan att det kanske inte är databasen det är fel på utan kanske hur DU använder den.

## 4. PHP & Säkerhet.

### 1. . mysql\_escape\_string() eller mysql\_real\_escape\_string()

går att använda som jag gjorde i koden:

```
$username = $connect -> real_escape_string($_POST['användarnamn']);
```

```
$password = $connect -> real_escape_string($_POST['lösenord']);
```

```
$firstname = $connect -> real_escape_string($_POST['förnamn']);
```

```
$secondname = $connect -> real_escape_string($_POST['efternamn']);
```

```
$mobilnr = $connect -> real_escape_string($_POST['teleniummer']);
```

```
$postnr = $connect -> real_escape_string($_POST['postnummer']);
```

```
$epost = $connect -> real_escape_string($_POST['email']);
```

## 2. Password\_hash() och Password\_verify()

password\_hash() är en ny funktion i senare PHP versioner där du inte behöver kryptera till specifik hashkod så som Sha1, md5 osv utan password\_hash() krypterar åt en och password\_verify() verifierar koden på nästa hemsida eller angiven hashkod. så när man hämtar en viss rad ur databasen med password\_hash kod så kommer det omvandla det till ordinarie inmatad sträng/int nr. password\_hash() gör en olik genererad kod varje gång den matas in och samma lösenord får få olika hash koder.

Algoritmerna som finns till password\_hash() är PASSWORD\_DEFAULT, PASSWORD\_BCRYPT, PASSWORD\_ARGON2I, PASSWORD\_ARGON2ID

tillhörande funktion är:

password\_algos(): array som ger möjlighet att lista algoritmerna som används.

password\_get\_info — ger information om hashen.

password\_needs\_rehash(string \$hash, string|int|null \$algo, array \$options = []): bool - rehashar.

## 3. säkrare

include

<http://www.phpportalen.net/wiki/index.php?page=PHP+och+s%E4kerhet>

<?php

```
if (!empty($_GET['sida'])) {
```

```
    $sida = substr($_GET['sida'], strrpos($_GET['sida'], '/')+1);
```

```
    if (is_readable($sida)) {
```

```
        include $_GET['sida'].'.php';
```

```
}
```

```
}
```

```
?>
```

gäller även för

[copy()]

[file()]

[fopen()]

[highlight\_file()]

[include()]

[move\_uploaded\_file()]

[popen()]

[readfile()]

[require()]

[show\_source()]

[unlink()]

#### **4. Cross-Site scripting (XSS)**

HTML special chars & ENT\_QUOTES

**5.** always use SSL

**6.** alltid ha uppdaterad PHP version.

**7.** SQL injection - <https://www.freelancer.com/community/articles/8-very-important-php-security-tips-you-should-follow>

Use libraries such as MySQLi Extension (MySQL Improved) or PDO that help you with statements that have bound variables.

Delete all idle procedures from your site.

Ensure you perform data validation and confirm that every input from users has the anticipated data type.

Avoid usage of super user access to connect to your site. Always have customized users, and limit their privileges.

Ensure that you strongly type query APIs with parameters, and with some substitution markers.

## **8. Cross Site Request Forgery (CSRF) attacks -** <https://www.freelancer.com/community/articles/8-very-important-php-security-tips-you-should-follow>

Use open libraries such as NoCSRF, CSRF Protection, anticsurf, and Clfsrpm.net to protect your website from CSRF susceptibility.

Use Unpredictable Synchronizer Token Pattern. This method helps a website to generate a random token which associates with the current session of a user as a hidden value. Once the site submits the form, your site can verify if the random token has come through a request, and if it is right.

Use Completely Automated Public Turing Test to tell Computers and Humans Apart (CAPTCHA) verification in forms to avoid CSRF. CAPTCHA that comes from the side of the client is legitimate since no hacker can guess its pattern. However, this gives the user another burden, and it can destroy the overall web experience of the user.

The application must ascertain the referral header. The website should block any request that comes from another domain. You can do away with vulnerability if the site allows requests from a similar domain. If there are HTTPS connections, this method fails since it omits the referrer.

9. Understand that web application security is an endless process. All PHP developers should update themselves about all new forms of exploits, and new flaws. Ensure you do not trust user input, and guard your database and file system. Visit CGISecurity.Net, PHP Manual Security Section, OWASP, and The PHP Security Consortium for more guidance about PHP security.
10. Finns flera sätt att hitta information om ny exploits med mera för specifika versioner med mera och sedan få en idé om hur man kan förebygga detta.

## 12. Nytt GANTT-Schema

[https://docs.google.com/spreadsheets/d/1tPT04RG5nxf5xbHQ1gZIs7M\\_Cu25Sp1LOTRRqNwprdg/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1tPT04RG5nxf5xbHQ1gZIs7M_Cu25Sp1LOTRRqNwprdg/edit?usp=sharing)

Skickat en länk som ska vara öppen för alla med länken och blir en liten revidering när Modul 5 är klar.

### GANTT Schema

| PROJECT TITLE   |                                      | Källan Bio: Webbserver programmering 2 |            |          |          |                      | SCHOOL NAME     |    | Hemods  |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
|-----------------|--------------------------------------|--|------------|----------|----------|----------------------|-----------------|----|---------|----|----|----|-----------------|----|----|----|----|----|-----------------|----|----|----|----|----|----------------|----|----|----|---|---|---------------|---|---|---|--|--|
| PROJECT MANAGER |                                      | Kevin Swärdh                           |            |          |          |                      | DATE            |    | 30/1/22 |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| WBS NUMBER      | TASK TITLE                           | TASK OWNER                             | START DATE | DUE DATE | DURATION | PCT OF TASK COMPLETE | Period 1        |    |         |    |    |    | Period 2        |    |    |    |    |    | Period 3        |    |    |    |    |    | Period 4       |    |    |    |   |   | Period 5      |   |   |   |  |  |
|                 |                                      |  |            |          |          |                      | 10 Jan - 15 Jan |    |         |    |    |    | 15 Jan - 21 Jan |    |    |    |    |    | 21 Jan - 26 Jan |    |    |    |    |    | 26 Jan - 1 Feb |    |    |    |   |   | 1 Jan - 6 Feb |   |   |   |  |  |
|                 |                                      |  |            |          |          |                      | 10              | 11 | 12      | 13 | 14 | 15 | 16              | 17 | 18 | 19 | 20 | 21 | 22              | 23 | 24 | 25 | 26 | 27 | 28             | 29 | 30 | 31 | 1 | 2 | 3             | 4 | 5 | 6 |  |  |
| 1               | Modul 1                              |  |            |          |          |                      |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 1.1             | Ett IT-Projekt                       | Kevin S.                               | 1/10/22    | 1/11/22  | 1        | 100%                 |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 1.1.1           | PHP-Projekt                          | Kevin S.                               | 1/11/22    | 1/13/22  | 2        | 100%                 |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 1.2             | SQL                                  | Kevin S.                               | 1/11/22    | 1/14/22  | 3        | 100%                 |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 1.3             | Uppdrag 1                            | Kevin S.                               | 1/15/22    | 1/15/22  | 0        | 100%                 |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 2               | Modul 2                              |  |            |          |          |                      |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 2.1             | PHP och hantering av data och inlogg | Kevin S.                               | 1/10/22    | 1/10/22  | 4        | 100%                 |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 2.2             | Uppdrag 2                            | Kevin S.                               | 1/10/22    | 1/10/22  | 3        | 100%                 |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 3               | Modul 3                              |  |            |          |          |                      |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 3.1             | AJAX & json med JS                   | Kevin S.                               | 1/10/22    | 1/10/22  | 0        | 100%                 |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 3.2             | AJAX & json med PHP                  | Kevin S.                               | 1/10/22    | 1/10/22  | 0        | 100%                 |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 3.2.1           | Uppdrag 3                            | Kevin S.                               | 1/10/22    | 1/10/22  | 0        | 100%                 |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 4               | Modul 4                              |  |            |          |          |                      |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 4.1             | Säkerhet & Optimering                | Kevin S.                               | 1/10/22    | 1/10/22  | 0        | 100%                 |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 4.2             | Uppdrag 4                            | Kevin S.                               | 1/10/22    | 1/10/22  | 0        | 99%                  |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |
| 5               | Modul 5                              |  |            |          |          |                      |                 |    |         |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |                |    |    |    |   |   |               |   |   |   |  |  |

## 13. Sammanfattning

Känner att efter denna kurs när jag kollar tillbaka på det hela inser man hur mycket planeringen faktiskt har betydelse i arbetet, hur allting blir mer organiserat tsm med alla olika personer som är involverade oavsett hur upplägget är uppbyggt. Planering upplägg och sedan revidering under arbetets resa är något som får arbetet att utvecklas och man har något fram för sig som man kan se vad man gjort och inte gjort. Man kan lättare med GANTT-schema veta om man håller sig till tidsramen eller om man måste lägga till, man ser lättare vilka som gör vad och vad de redan gjort. En projekt-plan och kravspec med mera, Ger möjligheten att alltid gå tillbaka och kolla på vad huvud målet var när man kört fast i kreativiteten eller dylikt. Ut över det känner jag att jag fick rätt på en hel del grejer, fick förståelse av rätt många saker jag hade funderingar över och tyckte det var roligt att få grunder angående AJAX & JSON som är effektiva användningsätt på en hemsida då den känns mer levande eller funktionell. Säkerheten är något jag kommer jobba mer med och känns lite som att det är som jag läste att säkerhet är ständigt utvecklande efter vad som händer, hur det händer och när det händer, nya saker upptäcks, andra försvinner, vissa hot är ständiga med mera. Tror man ska få en mer övergripande förståelse angående säkerhet för att lättare lägga till och ta bort efter behov vilket jag då tror är bra att börja i linux då det krävs mer åtgärder men att det fortfarande finns färdiga program.

I helhet är jag väldigt nöjd med min Biograf, fått inlogg för både admin/kund att fungera, där personer får egna sidor, kopplat sessions id där det printar deras namn på deras Mina Sidor, totalt pris uppdaterar sig utan att hemsidan behöver uppdateras och mycket kvar, men det är jag i ett nötskal också tror jag att jag får ständigt nya idéer på vad som ska göras och vill göra allt. Men fått en bra grund och projektet kommer jag fin slipa och uppdatera allt efter som när jag har tid framåt. Det jag t.ex lägger till beställning, och där den sedan visas på enskild persons sida, en tabell på admin sidan med alla beställningar, email funktioner, kategorier på filmer sidan med flytande divar och möjligheten att ändra sina uppgifter igenom knapp tryck med alter eller update där det blir en ny popup vid varje ändra knapp. Mycket att kunna tillägga, mycket lärt. Men är rätt nöjd med detta hela och fått en rätt stor uppfattning över ett IT/PHP projekt i helhet och hur allting kan gå till med mera.

## 14. Innehållsförteckning

- 1. Inledning
- 2. Syfte
- 3. Händelseförlopp
  - 3.3. Hemsidans kod
  - 3.4. Hemsidans SQL
- 4. Slutprodukten
- 5. Övrig information
- 6. Planering & GANTT-Schema
- 7. Uppdaterad layout
- 8. Admin oklar & PHP texter
- 9. Komplet admin/member
  - 9.1 Inloggningsystem
- 10. AJAX & PHP/JSON
- 11. Säkerhet & Optimering



11.1 Apache, Optimering & Säkerhet

11.2 Mysql Optimering

11.3 UTvbloggen

11.4 PHP & Säkerhet

12. Nytt GANTT-Schema

13. Sammanfattning