

DATABASHANTERING (UPPDATERA)

STEG 1

Skapa filen **edit.php** i huvudmappen. I denna fil ska det grafiska gränssnittet för uppdatering utvecklas i form av ett formulär. Samtidigt behövs funktionalitet för att uppdatera information. Skapa därför filen **update.php** i mappen **assets/functions/**.

Utöver detta behöver vi en funktion för att hämta information om en enskild användare. Skapa därför filen **select-id.php** i mappen **assets/functions/**.

Nya filer är **fetmarkerade**.

- assets/
 - o config/
 - db.php
 - o css/
 - album.css
 - all.min.css
 - bootstrap.min.css
 - o functions/
 - insert.php
 - **select-id.php**
 - select.php
 - **update.php**
 - o images/
 - o webfonts/
- add.php
- **edit.php**
- index.php

STEG 2

Öppna **index.php** och **koppla ett id-nummer** till respektive användare i while-loopen. Frågetecken anger att en GET-variabel ska skapas. GET-variabeln kallar vi i detta exempel för **id**. Ange först ett tabellhuvud på följande sätt:

- `<th colspan="2">Administration</th>`

Ange därefter i while-loopen två dataceller på följande sätt:

- `<td>Uppdatera</td>`
- `<td>Radera</td>`

STEG 3

- Högerklicka på filen **add.php** och välj **Clone**.
- Ändra namnet på den nya filen **edit.php**.
- Öppna filen **edit.php**.

Överst på edit.php ska det inkluderas funktionalitet för uppdatering av användare och för att hämta information om en enskild användare. Ange därför följande:

```
<?php
// Opens database connection
require_once 'assets/config/db.php';
// Update information to database
require_once 'assets/functions/update.php';
// Get specific information about user
require_once 'assets/functions/select-id.php';
?>
```

STEG 4

Processen för att uppdatera är enligt följande:

1. Hämta information om enskild användare (**select-id.php**).
2. Presentera denna i ett formulär (**edit.php**).
3. Erbjud möjligheten att uppdatera informationen (**update.php**).

Börja med att utveckla funktionalitet för att hämta information om en enskild användare. Öppna filen **select-id.php** och ange följande:

```
<?php
// Checks whether ID is available as GET
if (isset($_GET['id'])) {
    // Gets specific post from database
    $sql = 'SELECT * FROM users WHERE id = :id';
    // Prepares a query
    $stmt = $dbh->prepare($sql);
    // Connects GET-variable with db containers
    $stmt->bindValue(':id', $_GET['id']);
    // Sends query to database
    $stmt->execute();
    // Adds all information about user to variable
    $row = $stmt->fetch();
}
?>
```

STEG 5

Nu är det dags att presentera denna i ett formulär. Öppna filen **edit.php** och börja med att **ändra <form> elementet** så att den pekar mot **edit.php** enligt följande:

- `<form action="edit.php" method="post">`

Ange därefter den hämtade informationen (**\$row**) i attributet **value** för varje formulärfält enligt följande:

```
<div class="form-group">
  <label for="firstname">Förnamn</label>
  <input type="text" class="form-control col-5" id="firstname" name="firstname"
value="<?php echo $row['firstname']; ?>">
</div>
<div class="form-group">
  <label for="lastname">Efternamn</label>
  <input type="text" class="form-control col-5" id="lastname" name="lastname"
value="<?php echo $row['lastname']; ?>">
</div>
<div class="form-group">
  <label for="email">E-post</label>
  <input type="text" class="form-control col-5" id="email" name="email" value="<?php
echo $row['email']; ?>">
</div>
```

STEG 6

Ändra namnet för knappen från **register** till **modify** samt texten från **Registrera** till **Uppdatera** enligt följande:

```
<button type="submit" class="btn btn-primary" name="modify">
  <i class="fas fa-pencil-alt"></i> Uppdatera
</button>
```

STEG 7

När vi skickar detta formuläret så måste vi säga till databasen specifikt vem vi ska uppdatera databasen för. Ett sätt är att skicka id-numret för användaren **gömt** i ett formulärfält.

Detta görs genom att lägga till följande direkt innan **</form>**:

- `<input type="hidden" name="id" value="<?php echo $row['id']; ?>">`

Testa nu att gå till `index.php` och klicka på en uppdatera-länk. Informationen för en användare ska nu visas i formuläret.

Högerklicka på sidan `edit.php`, välj visa källkod och bekräfta att id-numret visas i slutet av formuläret.

STEG 8

Öppna filen **update.php** för att ange funktionalitet för att uppdatera information till databasen.

```
<?php
// Checks whether the update button has been pressed
if (isset($_POST['modify'])) {
    // Creates a query
    $sql = '
        UPDATE users
        SET firstname = :firstname,
            lastname = :lastname,
            email = :email
        WHERE id = :id
    ';
    // Prepares a query
    $stmt = $dbh->prepare($sql);
    // Connects form fields with db containers
    $stmt->bindValue(':firstname', $_POST['firstname']);
    $stmt->bindValue(':lastname', $_POST['lastname']);
    $stmt->bindValue(':email', $_POST['email']);
    $stmt->bindValue(':id', $_POST['id']);
    // Sends query to database
    if ($stmt->execute()) {
        header('Location: ../../index.php?action=updated');
        exit();
    }
}
?>
```

STEG 9

Avsluta genom att skriva ut en **alert-ruta** på **index.php** som anger feedback för uppdatering. Lägga följande i **switch-satsen** från tidigare genomgång:

```
case 'updated':  
    echo '  
        <div class="alert alert-success">  
            Posten har uppdaterats i databasen!  
        </div>  
    ';  
    break;
```

Testa funktionen så att den fungerar i sin helhet.