

Backend Selektion=5;

Utbildare: Mahmud Al Hakim

NACKADEMIN

Lektionstillfällets mål

- Applikationsutveckling med PHP och MySQL – del 1 av 2
- *Att läsa: sid. 152–199*

Kort summering av föregående lektion

- Vi har gått igenom hur man utvecklar en WordPress Sajt/blogg.

Vi ska utveckla en enkel webbapplikation

Mahmuds Telefonlista

Ange namn

Ange telefon/mobil

Lägg till

Namn

Telefon

Admin

Mahmud

08-12345678

Uppdatera | Ta bort

Kalle

070-1234567

Uppdatera | Ta bort

Kravspecifikation

- **Utveckla en MySQL-databas.**
 - Databasnamnet är **contactsdb**.
 - Databasen har en enda tabell, **contacts**
 - Tabellen har tre fält: **id** int(11), **name** varchar(50) och **tel** varchar(50).
- **Utveckla en CRUD-Applikation.**
 - **Create**
Man ska kunna lägga till namn och telefonnummer i databasen.
 - **Read**
Man ska kunna visa en lista/tabell över alla kontakter.
 - **Update**
Man ska kunna uppdatera namn och/eller telefonnummer.
 - **Delete**
Man ska kunna ta bort en kontakt.

Arbetsprocessen

1. Skapa en databas via phpMyAdmin
2. Skapa en startsida med hjälp av Bootstrap starter template.
3. Databaskoppling med PDO.
4. Arbeta med Create.
5. Arbeta med Read.
6. Arbeta med Update.
7. Arbeta med Delete.

Steg 1: Skapa en databas via phpMyAdmin

The screenshot shows the phpMyAdmin interface for a MySQL database. The top navigation bar indicates the connection is to 'Server: localhost:3306' and the current database is 'Database: contactsdb'. The table being viewed is 'Table: contacts'. Below the navigation bar is a toolbar with links for 'Browse', 'Structure' (which is currently selected), 'SQL', 'Search', 'Insert', 'Export', and 'Import'. Underneath the toolbar, there are two tabs: 'Table structure' (selected) and 'Relation view'. The main content area displays the structure of the 'contacts' table with the following columns:

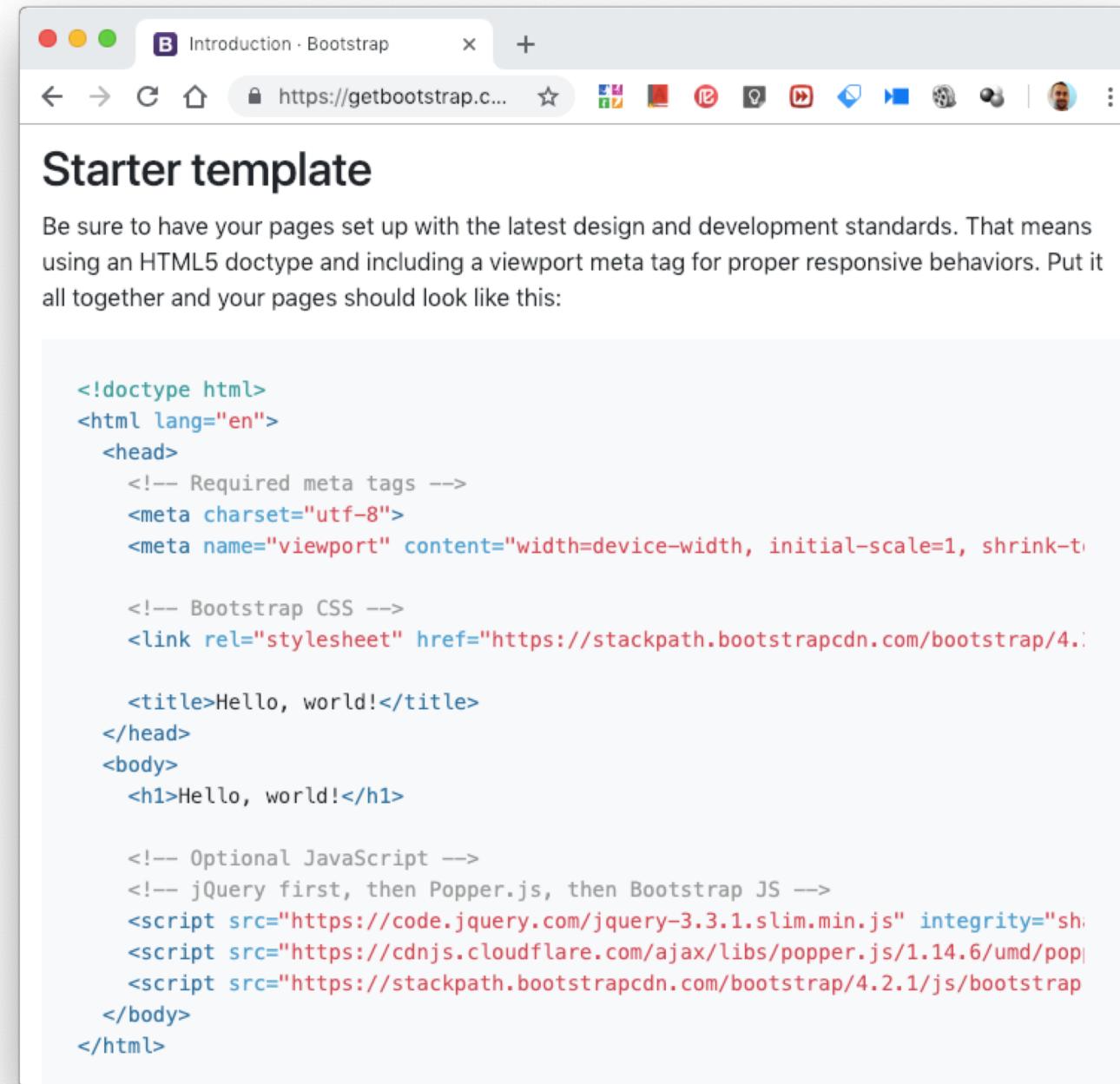
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	name	varchar(50)	utf8_unicode_ci		No	None		
3	tel	varchar(50)	utf8_unicode_ci		No	None		

Steg 2

Skapa en startsida med hjälp av Bootstrap starter template

Skapa filen **index.php**

Gå till <https://getbootstrap.com> och hämta en starter template.



The screenshot shows a web browser window with the title 'Introduction · Bootstrap'. The URL in the address bar is 'https://getbootstrap.com'. The page content is titled 'Starter template' and contains instructions: 'Be sure to have your pages set up with the latest design and development standards. That means using an HTML5 doctype and including a viewport meta tag for proper responsive behaviors. Put it all together and your pages should look like this:' followed by a block of HTML code.

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=1, viewport-fit=cover">
    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-qJjUZ5NQeVII+qiM/eB+4u1Kqyv8P+X1rY0Cgk3+pqLWZ+qD8cLwzqDqEJZ" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.6/dist/umd/popper.min.js" integrity="sha384-wSHIlk-eyHLqVwR8eC8XzZ8AxHZHdewo8DQaGtD5z8mN9AlHg1QK85K6jZ" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.2.1/dist/js/bootstrap.min.js" integrity="sha384-B0UglyR+jN6PgENGvQ1UmC0C78R7G+Hj3JPUK1BkqM0nQ4aFw7dZgGAJHg" crossorigin="anonymous"></script>
  </body>
</html>
```

Olika sätt att hantera databaser med PHP

MySQL

(OBS! deprecated)

MySQLi

(Improved Extension)

PDO

(PHP Data Objects)



Should I Use MySQLi or PDO?
If you need a short answer, it would be
"Whatever you like".

https://www.w3schools.com/php7/php7_mysql_connect.asp

Steg 3 – Databasuppkoppling med PDO

```
$server      = "localhost";
$username   = "root";
$password   = "root";
$database   = "contactsdb";

try{
    $conn = new PDO("mysql:host=$server;dbname=$database", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
}

catch(PDOException $e){
    echo $e-> getMessage();
}
```

Include och require

- Du kan inkludera skript-filer som används av flera dokument med hjälp av include-satsen, t.ex. sidhuvud och sidfot.
- Om skriptet är viktigt för din applikation så använd istället require-satsen. Då visar PHP ett felmeddelande om skriptet saknas!

Use `require` when the file is required by the application.

Use `include` when the file is not required and application should continue when file is not found.

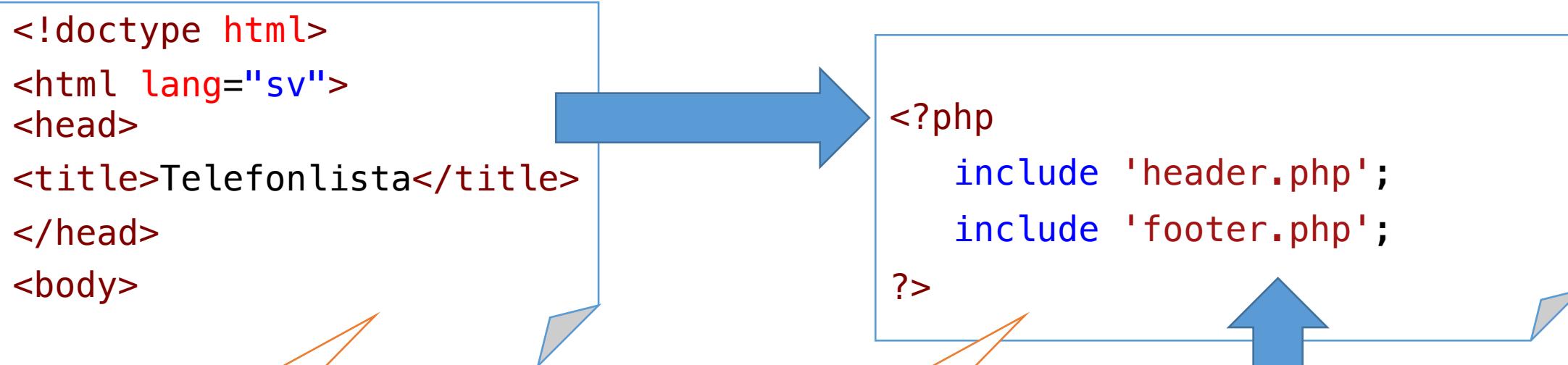
PHP include – Exempel

```
<!doctype html>
<html lang="sv">
<head>
<title>Telefonlista</title>
</head>
<body>
```

header.php

footer.php

```
<hr>
<footer>Copyright &copy; <?php echo date('Y'); ?></footer>
</body>
</html>
```



PHP require – Exempel

```
<?php  
$server      = "localhost";  
$username   = "root";  
$password   = "root";  
$database   = "contactsdb";  
  
$conn = new  
PDO("mysql:host=$server;dbname=$database",  
$username, $password);
```

```
<?php  
require 'db.php';
```

Filén
db.php

Filén
index.php

Steg 4 – Arbeta med Create. Skapa ett formulär

```
<form action="#" method="post">  
<div class="form-row">  
  <div class="col-md-4">  
    <input type="text" name="name" class="form-control mt-2"  
          placeholder="Ange namn">  
  </div>  
  <div class="col-md-4">  
    <input type="text" name="tel" class="form-control mt-2"  
          placeholder="Ange telefon/mobil">  
  </div>  
  <div class="col-md-4">  
    <input type="submit" class="form-control mt-2 btn btn-outline-primary"  
          value="Lägg till" >  
  </div>  
</div>  
</form>
```

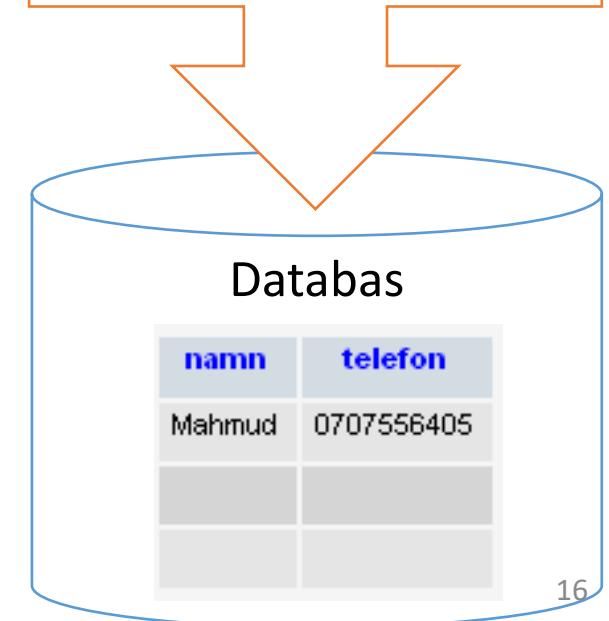
Skapa ett PHP-skript som hanterar data

Vid submit paketeras all data med hjälp av metoden post och skickas vidare till php-skriptet som finns i samma fil.

`action="#" method="post"`

PHP-Skript loggar in i databasen och skickar data via SQL

```
<?php  
  
if ($_SERVER['REQUEST_METHOD'] === 'POST'):  
    print_r($_POST);  
endif;  
  
?>
```



Lägg till en ny kontakt i databasen

```
// Förbered SQL-satsen and binda parametrar
$stmt = $conn->prepare("INSERT INTO contacts (name, tel)
                        VALUES (:name, :tel)");
$stmt->bindParam(':name', $name);
$stmt->bindParam(':tel' , $tel);

// Hämta data via post-arrayen
$name = $_POST['name'];
$tel  = $_POST['tel'];

// Kör SQL-satsen (infoga en rad)
$stmt->execute();
```

Steg 5 – Arbeta med Read

Hämta alla kontakter från databasen

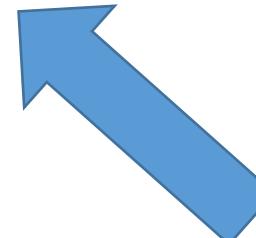
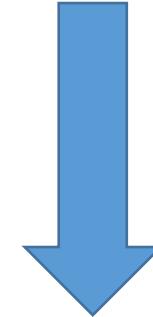
```
$stmt = $conn->prepare("SELECT * FROM contacts");
$stmt->execute();

$table = '<table class="table">';
$table .= '<tr><th>Namn</th><th>Telefon</th></tr>';
while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
    $table .= '<tr>';
    $table .= '<td>' . $row['name'] . '</td>';
    $table .= '<td>' . $row['tel'] . '</td>';
    $table .= '</tr>';
}
$table .= '</table>';
echo $table;
```

Steg 6 – Arbeta med Update

Lägg till en kolumn i tabellen

```
$table = '<table class="table">';  
$table .= '<tr><th>Namn</th><th>Telefon</th><th>Admin</th></tr>';  
while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {  
    $table .= '<tr>';  
    $table .= '<td>' . $row['name'] . '</td>';  
    $table .= '<td>' . $row['tel'] . '</td>';  
    $table .= '<td><a href="update.php?id=' . $row['id'] . '">  
        Uppdatera</a>  
    </td>';  
    $table .= '</tr>';  
}  
$table .= '</table>';
```



Filen update.php

```
// Gå till index.php om id saknas i URLen
// Gör detta högst upp i filen

if(empty($_GET['id'])){
    header('Location: index.php');
}
$id = $_GET['id'];

print_r($_GET);
```

Hämta data om aktuell kontakt från databasen

```
$id = $_GET['id'];
require 'db.php';
$stmt = $conn->prepare("SELECT * FROM contacts WHERE id=:id");
$stmt->bindParam(':id', $id);
$stmt->execute();
if($stmt->rowCount() > 0){
    $row = $stmt->fetch(PDO::FETCH_ASSOC);
    $name = $row['name'];
    $tel = $row['tel'];
}
else{
    header('Location: index.php');
}
```

Skapa ett uppdateringsformulär



```
<form method="post" action="?id=<?php echo $id; ?>">
<div class="form-row">
<div class="col-md-4">
<input type="text" name="name" value="<?php echo $name; ?>"
       class="form-control mt-2" placeholder="Ange namn">
</div>
<div class="col-md-4">
<input type="text" name="tel" value="<?php echo $tel; ?>"
       class="form-control mt-2" placeholder="Ange telefon/mobil">
</div>
<div class="col-md-4">
<input type="submit" class="form-control mt-2 btn btn-outline-primary"
       value="Uppdatera" >
</div>
</div>
</form>
```



Skicka uppdateringen till databasen

```
if ($_SERVER['REQUEST_METHOD'] === 'POST'):

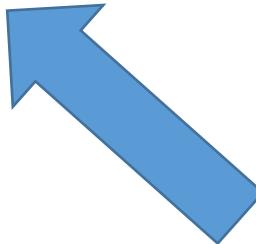
$name = $_POST['name'];
$tel = $_POST['tel'];
$id = $_GET['id'];
$sql = "UPDATE contacts SET name = :name, tel = :tel WHERE id = :id";
$stmt = $conn->prepare($sql);
$stmt->bindParam(':name', $name);
$stmt->bindParam(':tel', $tel);
$stmt->bindParam(':id', $id);
$stmt->execute();

endif;
```

Steg 7 – Arbeta med Delete

Lägg till en länk i tabellen

```
$table .= '<td>
    <a href="update.php?id=' . $row['id'] . '">
        Uppdatera
    </a>
    |
    <a href="delete.php?id=' . $row['id'] . '">
        Ta bort
    </a>
</td>';
```



Ta bort en rad från databasen

```
$id = $_GET['id'];
$sql = "DELETE FROM contacts WHERE id = :id";
$stmt = $conn->prepare($sql);
$stmt->bindParam(':id', $id);
$stmt->execute();
header('Location: index.php');
```

Övning

- Skapa ett kontaktformulär med följande fält
 - Namn
 - E-post
 - Meddelande
- Alla fält är obligatoriska.
- Skicka alla meddelanden till en databas.
- Visa alla meddelanden i en HTML-tabell.
- *Läs: sid. 152–199 i boken.*

Summering av dagens lektion

- Vi har gått igenom grunderna i applikationsutveckling med PHP och MySQL
- Reflektioner kring dagens lektion?

Framåtblick inför nästa lektion

Under nästa lektion kommer vi att jobba mer med applikationsutveckling i PHP och MySQL.

NACKADEMIN