

3)Produkte aus DB anzeigen und Detailseite erstellen (Shop) inkl. Übergabe der ID mit der GET-Methode

a)card.php

Die Startseite des Shops greift zurzeit noch auf die Daten aus dem Ordner „assets“ zu. Damit kann man aber mit vielen Produkten nicht arbeiten. Daher muss die Versorgung der Daten aus der Datenbank erfolgen.

Öffne im Ordner „templates“ eine Datei „card.php“.

Reduziere diese auf eine „col“.

```
card.php
templates > card.php
1 <div class="container">
2   <div class="card">
3     <div class="card-title">weißer DigBiz</div>
4     <div class="card-body">
5       
6     </div>
7     <div class="card-footer">
8       <a href="#"><button class="btn btn-primary btn-sm">Detail</button></a>
9       <a href="#"><button class="btn btn-success btn-sm">In den Warenkorb</button></a>
10    </div>
11  </div>
12 </div>
13
```

Statt den fixen Einträgen sollen die Daten hier bald aus der Datenbank kommen. Daher werden diese nun ersetzt durch die PHP-echo Funktion.

```
<div class="card-title"><?php echo $row['name'] . ?></div>
```

Aber zur Vereinfachung kann man die Kurzschreibeweise für „echo“ verwenden, nämlich

```
<?= $row['name'] ?>
```

<?= \$row['name'] ?>

Zentrieren funktioniert mit: d-flex justify-content-center

Nutze das für den Titel und vergrößere diesen gleich mit <h4>:

```
card.php
<div class="col">
  <div class="card">
    <div class="card-title d-flex justify-content-center"><h4><?= $row['name'] ?></h4></div>
    <div class="card-body">
```

Im <body> soll zuerst das Bild, dann der Preis angezeigt werden, beides aus der Datenbank geholt.

Das Bild aus der Datenbank holen:

Ändere dazu in der Zeile 5 den Pfad aus „assets“. BILD: bedarf eines speziellen Codes:

```
<?php echo
"<img height='200' width='auto' src='data:image/jpeg;base64, ".base64_encode($row['bild']). "'>"
?>
```

Für das Zentrieren muss man extra 2 <div> erstellen und zentrieren und das Bild sowie den Preis jeweils hinein stellen.

```
4 <div class="card-body">
5 <div class="d-flex justify-content-center"><!-- Inhalt zentrieren -->
6 <p>
7 <?php echo
8 "<img height='200' width='auto' src='data:image/jpeg;base64, ".base64_encode($row['bild'])." '>"
9 ?> <!-- Bild aus DB holen -->
10 </p>
11 </div>
12 <div class="d-flex justify-content-center">
13 <p><?=$row['preis'] ?></p>
14 </div>
15 </div>
```

Zur Info:

Die Bezeichnungen aus der Datenbank lauten: name – beschreibung – farbe – preis und bild

p_id	name	beschreibung	farbe	preis	bild
5	Aufklappmenüs anklippen	creme-weißer stylischer Hoodie, aus Baumwolle mit ...	weiß	39.90	[BLOB - 62,9 KiB]

Öffne die „shop.php“.

Für eine schöne Darstellung von jeweils 3 Cards pro Reihe (row) füge diese Information hier ein. Wenn man eventuell mehr Produkte nebeneinander anzeigen lassen möchte eben etwas abgewandelt.

In Zeile 28 und der dazu passenden Zeile 33, wo das <div> wieder geschlossen wird:

```
26
27 <div class="row">
28 <div class="col-md-4">
29 <?php
30     require "templates/card.php";
31 >?
32 </div>
33 </div> <!--ende row-->
34 </div>
35 </body>
36 </html>
```

Nun folgt die Anbindung an die Datenbank und die „while“-Schleife, die nacheinander die Produkte aus der Tabelle der Datenbank holt.

Füge ganz oben im PHP-Bereich die Verbindung ein:

- Zuerst die Beziehung zur Datenbank
- Dann die SQL-Abfrage aus der Tabelle mit dem Namen „produkte“ – auch wenn es aus Gründen der Sicherheit nicht optimal ist, alle Elemente mit einem Sternchen * zu holen, soll es hier so sein
- Dann eine Variable, in der die geholten Elemente gespeichert werden.

```

shop.php
1  <?php
2  error_reporting(-1);
3  ini_set('display_errors', 'On');
4
5  require "functions/database.php";
6
7  $sql = "SELECT * FROM produkte";
8  $result = $dbh->query($sql);
9  ?>
10
11 <!DOCTYPE html>

```

```

require "functions/database.php";
$sql = "SELECT * FROM produkte";
$result = $dbh->query($sql);

```

In der Klasse „row“, wo die geholten Resultate dann benötigt werden, muss nun die „while“-Schleife angeführt werden. Beachte, da diese von HTML unterbrochen wird, dass diese mit einem „Doppelpunkt“ endet bzw. nach der „col“ mit „endwhile“ schließt.

```

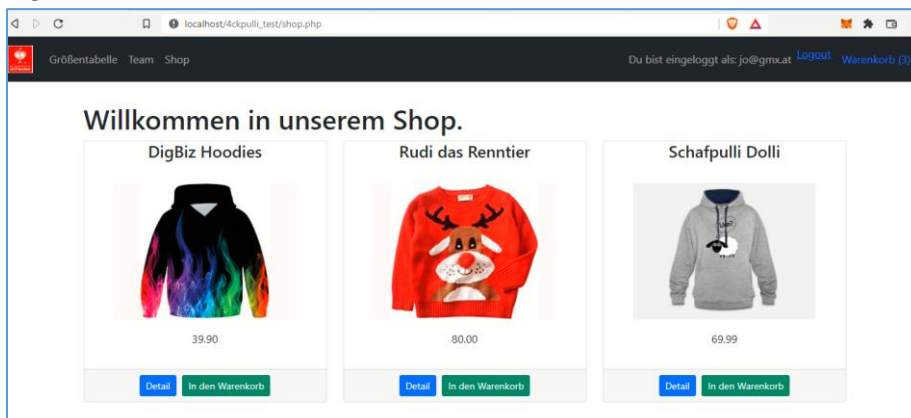
31 <div class="row">
32 <?php while($row = $result->fetch()):?> <!--hier der Doppelpunkt-->
33 <div class="col-md-4">
34 <?php
35     require "templates/card.php";
36 >?>
37 </div>
38 <?php endwhile;?>
39 </div> <!--ende row-->
40 </div>
41 </body>
42 </html>

```

```
<?php while($row = $result->fetch()):?> <!--hier der Doppelpunkt-->
```

```
<?php endwhile;?>
```

Ergebnis:



b) Neue Seite „details.php“

Detailseite mit der ID übergeben

Erstelle eine neue Seite „details.php“ im Ordner „templates“ und orientiere die dabei an der Seite „card.php“.

Kopiere den oberen Bereich bis zum Einbinden der „navbar.php“ aus der Datei „registrieren.php“.

```
templates > details.php
1 <!DOCTYPE html>
2 <html lang="de">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Pullishop</title>
8   <base href="/4ckpullis/">
9   <link rel="stylesheet" href="assets/css/bootstrap.css">
10  <link rel="stylesheet" href="assets/css/styles.css">
11 </head>
12 <body>
13 <?php
14   require "navbar.php";
15 >?>
16
17 <br><br><br><br>
```

Darunter soll nun die Darstellung wie folgt erscheinen:

Vorlage amazon.at:



Links das Bild – ca. 4 Spalten breit (col-4)

Rechts daneben der Titel, Beschreibung usw. – ca. 8 Spalten (col-8)

Erstelle eine „row“ mit „col-4“ und „col-8“.

Die col-4 nimmt das Bild auf. Kopiere das aus der gerade verwendeten Datei „card.php“

```

29
30 <br><br><br><br>
31 <div class="container">
32   <h1>Detailseite</h1>
33
34   <div class="row">
35     <div class="col-4">
36       <?php echo
37       <img height='200' width='auto' src='data:image/jpeg;base64, ".base64_encode($row['bild'])." '>
38     ?> <!-- Bild aus DB holen -->
39     </div>
40
41     <div class="col-8">
42     |
43     </div>
44   </div> <!-- ende row-->
45 </div> <!-- ende container-->
46
47 </body>
48 </html>

```

Die 8 Spalten orientieren sich an der oben erstellten „card“ aber ohne das Bild.

Jedoch soll der Button in Zeile 35 nicht „Detail“ heißen sondern „zurück“ und zum Shop zurückführen, damit der Kunde locker zurückkommt.

```

40
41   <div class="col-8">
42     <div class="container">
43       <div class="card">
44         <div class="card-title"><?=$row['name'] ?></div>
45         <div class="card-body">
46           <p><?=$row['farbe'] ?></p>
47           <p><?=$row['preis'] ?></p>
48         </div>
49         <div class="card-footer">
50           <a href="shop.php"><button class="btn btn-primary btn-sm">zurück</button></a>
51           <a href="#"><button class="btn btn-success btn-sm">In den Warenkorb</button></a>
52         </div>
53       </div>
54     </div>
55   </div> <!-- ende row-->
56 </div> <!-- ende container-->
57

```

Anbindung an die Datenbank auch hier ganz oben:

Kopiere das aus der Datei „shop.php“.

```

card.php  details.php X  shop.php
templates > details.php
1  <?php
2  error_reporting(-1);
3  ini_set('display_errors','On');
4
5  require "functions/database.php";
6
7  $sql = "SELECT * FROM produkte";
8  $result = $dbh->query($sql);
9  ?>
10
11 <!DOCTYPE html>
12 <html lang="de">

```

Es muss jedoch die Route zur Datenbank in Zeile 5 abgeändert werden, da die „details.php“ ganz wo anders liegt als die „shop.php“, nämlich:

```
5 require "../functions/database.php";
```

Jedoch benötigen wir hier keine „while“-Schleife sondern nur eine simple Ausgabe für das eine Ergebnis, welches in die Variable „\$row“ gespeichert wird. Diese Variable wird dann unten das passende Array-Element (z.B. \$row['name']) ausgegeben.

Füge nach der Zeile 8 ein diese Variable ein:

```
6
7 $sql = "SELECT * FROM produkte";
8 $result = $dbh->query($sql);
9 $row = $result->fetch();
10 ?>
```

Testen:

Nun erstellen wir zu **Probezwecken** eine Detailseite für das erste Produkt, nämlich im SQL-Befehl mit der Einschränkung

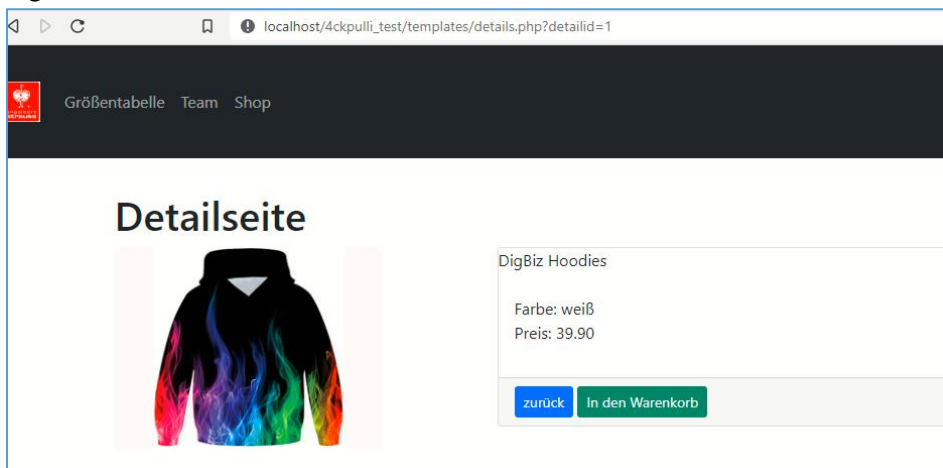
WHERE p_id = 1

```
5 require "../functions/database.php";
6
7 $sql = "SELECT * FROM produkte WHERE p_id = 1";
8 $result = $dbh->query($sql);
9 $row = $result->fetch();
10 ?>
```

Testen: Gib in die URL diese neue Seite ein:

<http://localhost/4ckpulli/templates/details.php>

Ergebnis:



Um nun automatisch auf diese Seite zu gelangen, muss man den Button in der Seite „shop.php“ noch adaptieren.

Verbindung von der Startseite des Shops „shop.php“ zur Detailseite erstellen

Öffne die Seite „card.php“, da ja hier dieser Button angezeigt wird, und nicht in der Seite „shop.php“, weil ja dort mittels „require“ auf die „card.php“ geleitet wurde.

Nun muss noch der Button geändert werden.

```
8     </div>
9     <div class="card-footer">
10        <a href="templates/details.php"><button class="btn btn-primary btn-sm">Detail</button></a>
11        <a href="#"><button class="btn btn-success btn-sm">In den Warenkorb</button></a>
12    </div>
```

Übergabe der ID mit Hilfe der GET-Methode – man muss ja wissen, welches Produkt (p_id) ausgewählt werden soll

Da in dieser Datei aber bereits die Elemente aus der Datenbank vorhanden sind, wie z.B. die „farbe“ und der „preis“, kann man auch sehr leicht auf die „p_id“ zugreifen.

Dafür genügt folgender Code, der am besten gleich an die URL angehängt wird, um sie so weiterzugeben. Damit kann die Seite mit den Details auch ganz bewusst auf die richtigen Details mit dieser ausgewählten ID geöffnet werden:

Dafür arbeitet man mit dem Fragezeichen. Dieses wird direkt an das Dateiformat angehängt und mit einem beliebigen Schlagwort versehen. Hier eignet sich gut „detailid“. Das wird von einem Istgleichzeichen gefolgt, welches auf die zu übergebende ID verweist:

Also:

- Fragezeichen
- beliebiges Schlagwort – wenn geht sinnvoll 😊
- Istgleichzeichen
- PHP-Echo mit der passenden ID aus der Datenbank

```
17     <div class="card-footer d-flex justify-content-center">
18        <a href="templates/details.php?detailid=?=$row['p_id']?>>
19        <button class="btn btn-primary btn-sm">Detail</button>
20    </a>
```

details.php

In der Seite „details.php“ kann man nun diese Information weiterverarbeiten.

- Zuerst übernimmt man die Info mittels der Superglobalen Funktion GET
- Wobei man diese Information in eine neue Variable speichert, wie hier
- Dann kann man diese Info in die SQL Abfrage einarbeiten, damit genau diese ID ausgegeben wird, die vorher angeklickt wurde

```

5  require "../functions/database.php";
6
7  $id = $_GET['detailid'];
8
9  $sql = "SELECT * FROM produkte WHERE p_id = 1";
10 $result = $dbh->query($sql);
11 $row = $result->fetch();
12 ?>

```

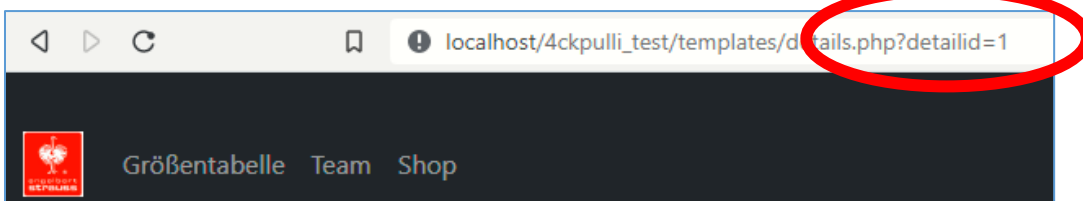
Diese neue Variable wird nun in der SQL-SELECT Methode mit dem Eintrag in der Datenbank verglichen. Ersetze daher den vorher fix eingestellten Datensatz „1“ mit dieser neuen Variablen:

```

7  $id = $_GET['detailid'];
8
9  $sql = "SELECT * FROM produkte WHERE p_id = $id";
10 $result = $dbh->query($sql);
11 $row = $result->fetch();

```

Ergebnis: in der URL wird die ID weitergegeben



Testen, ob alle Produkte auswählbar sind.

Fährt man mit dem Mauszeiger über den Button „Detail“ (ohne zu klicken) sieht man den Pfad, der zukünftigen URL. Hier kann man nun auch checken, ob man richtig gearbeitet hat, wenn ganz hinten die Nummer der ID steht.

