

---

# **Laborprotokoll**

## **WordPress Plug-In**

---

**INSY Übung  
4AHITM 2015/16**

**Kanyildiz Muhammedhizir**

**Note:**

**Betreuer: Prof. Brein**

**Version 1.0**

**Begonnen am 1. April 2016**

**Beendet am 6. April 2016**

## Inhaltsverzeichnis

1	Einführung.....	3
1.1	Ziele.....	3
1.2	Voraussetzungen.....	3
1.3	Aufgabenstellung.....	3
2	WordPress Installieren.....	4
2.1	MySQL DB + User erstellen.....	4
2.2	Wordpress herunterladen.....	4
3	Wordpress Anpassen.....	5
3.1	Die Wordpress Seite Öffnen.....	5
3.2	Eingloggen.....	7
4	Wordpress Plug-In Einbinden.....	7
4.1	Plug-In Verzeichnis.....	7
4.2	Plug-In in Wordpress Einbinden.....	8
5	Probleme.....	<b>Error! Bookmark not defined.</b>
6	Aufwandsabschätzung.....	9
7	Literatur Verzeichnisse.....	9
8	Quellen.....	<b>Error! Bookmark not defined.</b>

# 1 Einführung

## 1.1 Ziele

Es soll eine Wordpress-Seite aufgesetzt werden, die Daten aus der SchokoDB anzeigen kann, sowohl statisch als auch interaktiv abhängig von Benutzereingaben.

## 1.2 Voraussetzungen

Aufmerksames zuhören im Unterricht.

Verständnisvolles lesen.

Grundkenntnisse Wordpress

## 1.3 Aufgabenstellung

Installiere und konfiguriere eine Wordpress-Seite für unsere Schokofabrik. Die Seite muss keine besonderen statischen Inhalte aufweisen.

Entwickle ein Wordpress-Plug-In [1], dass du verwendest, das auf zwei Seiten eingebunden wird:

Schnäppchen sind entweder Kunstwerke mit einem Schätzwert unter 2000 Euro oder Süßigkeiten unter 3 Euro. Gib auf der Seite eine Liste von Schnäppchen (Bezeichnung und Gewicht) aus. Vermerke dabei auch in einer eigenen Spalte, ob es sich um ein Kunstwerk oder ein Produkt aus dem Standardsortiment handelt.

Auf einer anderen Seite soll der Benutzer die Möglichkeit haben, eine der in der Datenbank angelegten Kunstschaufen auszuwählen. Anschließend soll zur gewählten Kunstschau eine Liste mit allen ausgestellten Kunstwerken samt erreichten Plätzen angezeigt werden.

## 2 WordPress Installieren

### 2.1 MySQL DB + User erstellen

```
CREATE DATABASE Schokoladenfabrik;  
CREATE USER 'user01'@'%' IDENTIFIED BY 'user01';  
GRANT ALL PRIVILEGES ON * . * TO 'user01'@'%';
```

% ... .. Auf allen IP-Adressen zugänglich.

ON \* ... .. Auf alle Tabellen.

### 2.2 Wordpress herunterladen

Die Letzte Version von Wordpress herunterladen.

Downloads Ordner finden und reingehen.

Diesen Befehl dort ausführen:

```
wget http://wordpress.org/latest.tar.gz  
tar xzvf latest.tar.gz Entpacken:
```

In den HTML Ordner Moven:

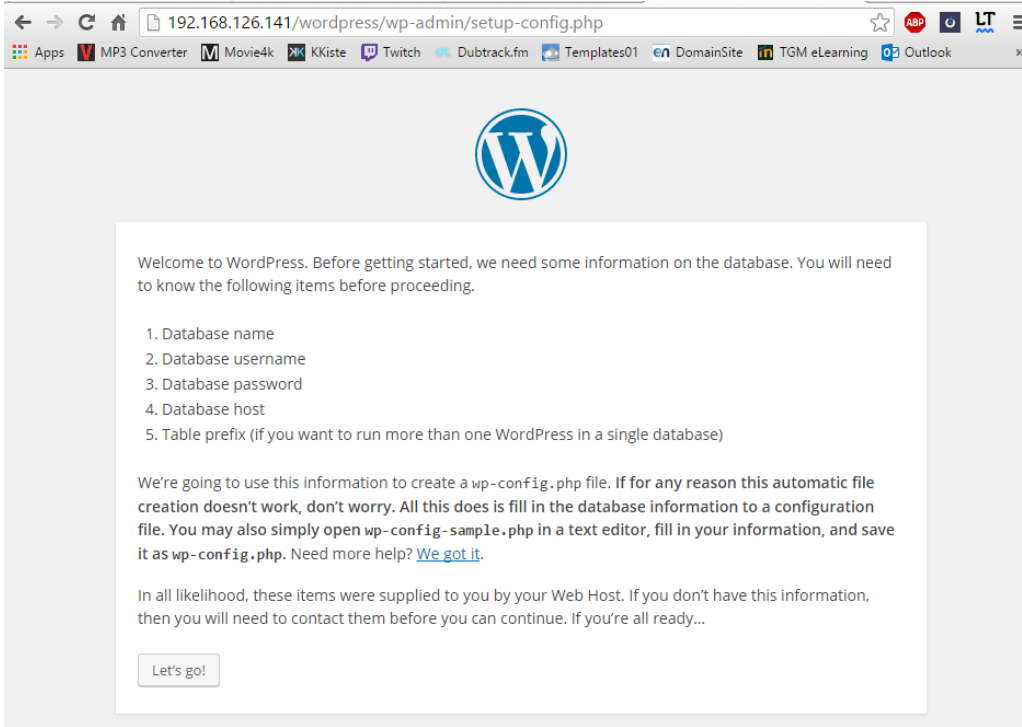
```
root@mkanyildiz:/home/mkanyildiz/Downloads# mv wordpress /var/www/html/
```

Rechte anpassen:

```
chmod -R 777 /var/www/html/*  
chmod -R 777 /var/www/html/wordpress/*
```

## 3 Wordpress Anpassen

### 3.1 Die Wordpress Seite Öffnen

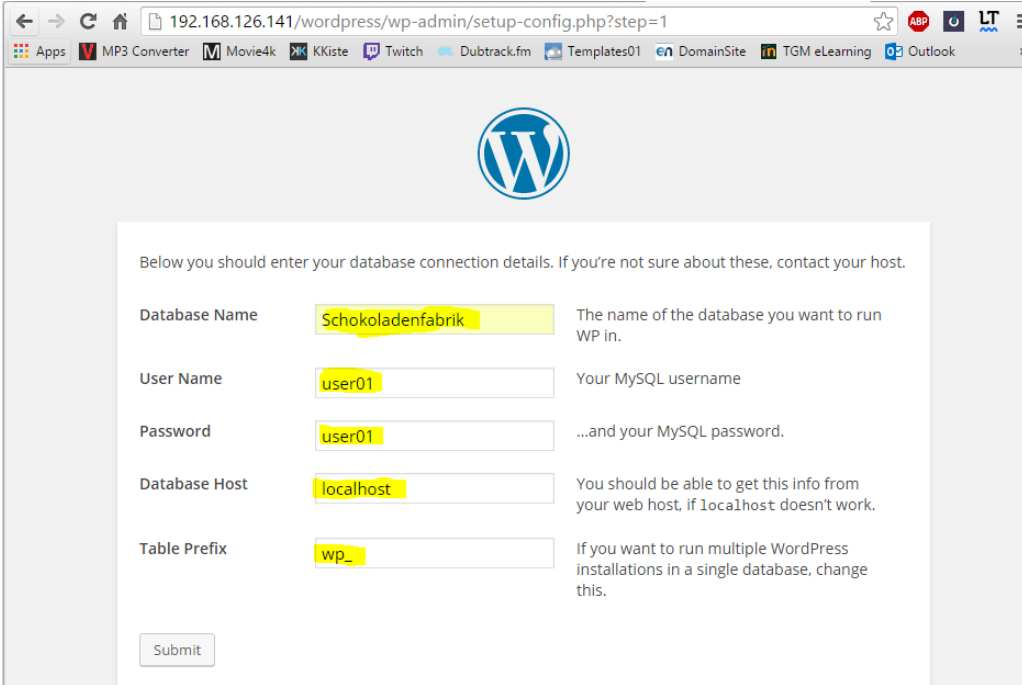


Welcome to WordPress. Before getting started, we need some information on the database. You will need to know the following items before proceeding.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)


We're going to use this information to create a `wp-config.php` file. If for any reason this automatic file creation doesn't work, don't worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`. Need more help? [We got it.](#)

In all likelihood, these items were supplied to you by your Web Host. If you don't have this information, then you will need to contact them before you can continue. If you're all ready...

Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="Schokoladenfabrik"/>	The name of the database you want to run WP in.
User Name	<input type="text" value="user01"/>	Your MySQL username
Password	<input type="text" value="user01"/>	...and your MySQL password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost doesn't work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.



## Welcome

Welcome to the famous five minute WordPress installation process! You may want to browse the [ReadMe documentation](#) at your leisure. Otherwise, just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

## Information needed

Please provide the following information. Don't worry, you can always change these settings later.

**Site Title**

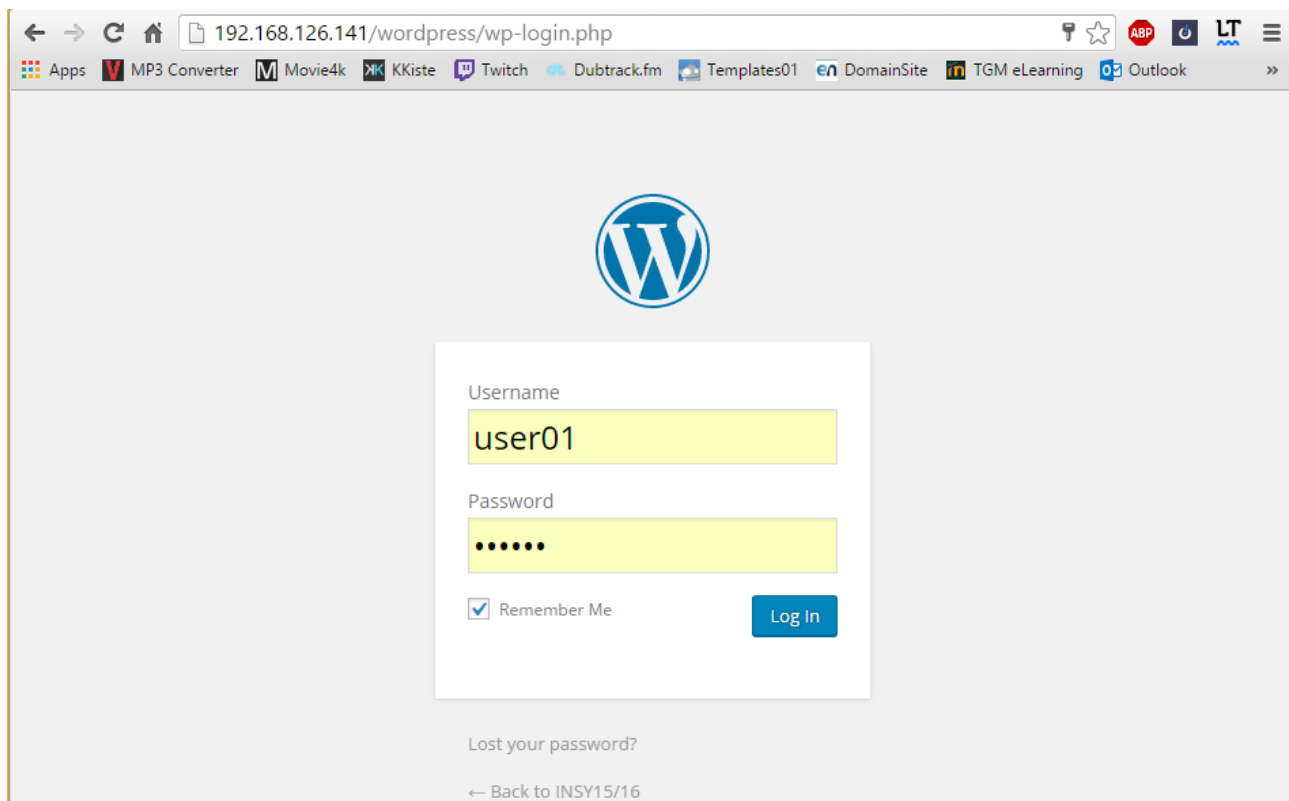
**Username**   
Username can have only alphanumeric characters, spaces, underscores, hyphens, periods and the @ symbol.

**Password, twice**  
A password will be automatically generated for you if you leave this blank.   
  
**Strength indicator**  
Hint: The password should be at least seven characters long. To make it stronger, use upper and lower case letters, numbers, and symbols like ! " ? \$ % ^ & }.

**Your E-mail**   
Double-check your email address before continuing.

**Privacy** ☒ Allow search engines to index this site.

## 3.2 Eingloggen



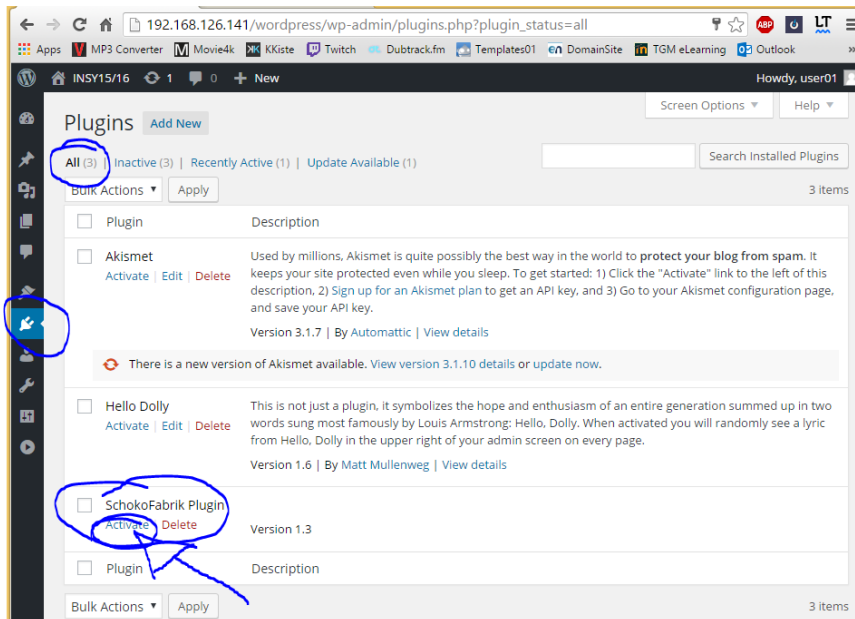
## 4 Wordpress Plug-In Einbinden

### 4.1 Plug-In Verzeichnis

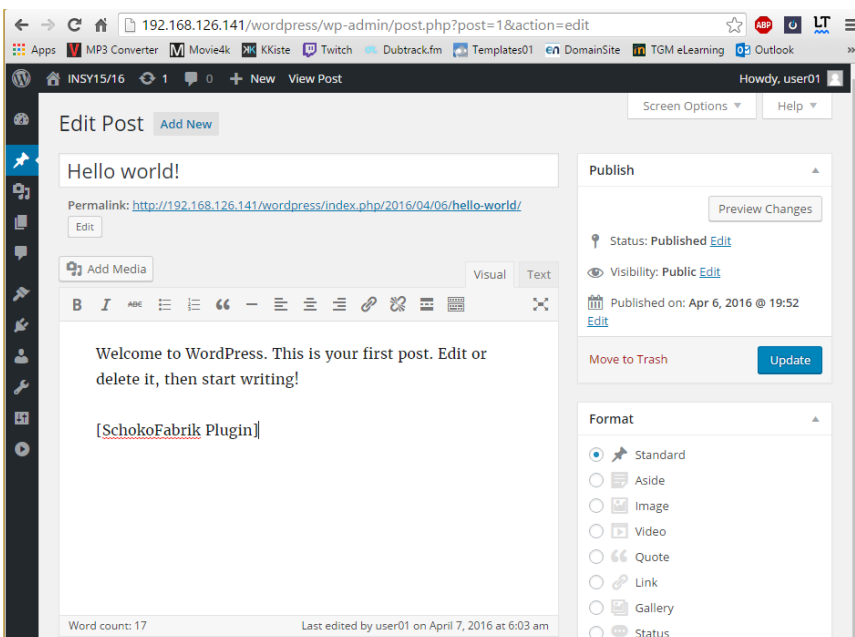
Im wordpress/WP-Content/Plug-Ins Ordner erstellen.  
z.b: Schokoladenfabrik/

/var/www/html/wordpress/wp-content/plugins/Schokoladenfabrik/ ✓			
Name	Size (KB)	Last modified	Owner
..			
Schokoladenfabrik.php	2	2016-04-07 ...	mkanyildiz

## 4.2 Plug-In in Wordpress Einbinden



Menu: Plug-Ins → All →  
Plug-In → Activate



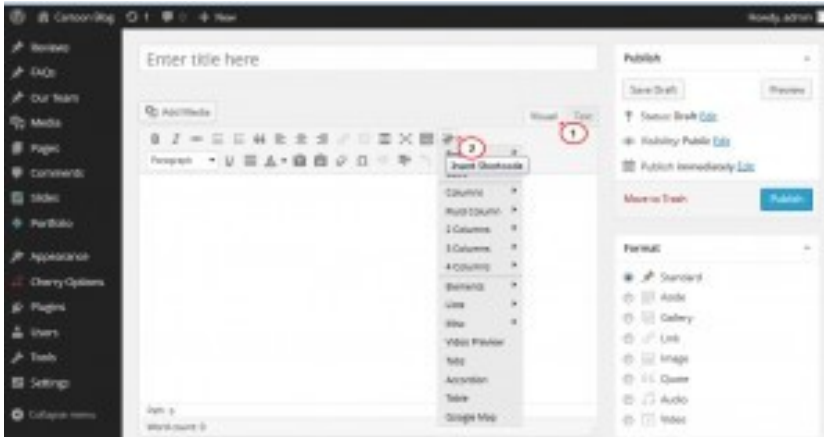
Menu: Beitrag → Beitrag  
Ändern → Plug-In  
Shortcut schreiben



## 5 Plug-In

Einträge/Seite im Admin-Panel WordPress öffnen.

Redigieren-Modus auf **Visual (Visuell)** umstellen. Den Button der Shortcodes drücken und notwendigen wählen.



## 6 Shortcode

### 6.1 Erklärung

Alle Shortcodes teilen sich bedingt in folgende Gruppen:

**Dynamische** – benutzt man, um den Content (Einträge, Typen übliche Einträge, Tags usw.) einzutasten;

**Spalten des Gitters** sind Spalten, um das Gitter zu erstellen;

**Elemente** sind zusätzliche Elemente des Contents;

**Andere** sind behelfsmäßige Elemente des erweiterten Content-Managements.

## 6.2 Beispiel

### Examples

Simplest example of a shortcode tag using the API: [footag foo="bar"]

```
function footag_func( $atts ) {  
    return "foo = {$atts['foo']}";  
}  
add_shortcode( 'footag', 'footag_func' );
```

Example with nice attribute defaults: [bartag foo="bar"]

```
function bartag_func( $atts ) {  
    $atts = shortcode_atts( array(  
        'foo' => 'no foo',  
        'baz' => 'default baz'  
    ), $atts, 'bartag' );  
  
    return "foo = {$atts['foo']}";  
}  
add_shortcode( 'bartag', 'bartag_func' );
```

Example with enclosed content: [baztag]content[/baztag]

```
function baztag_func( $atts, $content = "" ) {  
    return "content = $content";  
}  
add_shortcode( 'baztag', 'baztag_func' );
```

If your plugin is designed as a class write as follows:

```
class MyPlugin {  
    public static function baztag_func( $atts, $content = "" ) {  
        return "content = $content";  
    }  
}  
add_shortcode( 'baztag', array( 'MyPlugin', 'baztag_func' ) );
```

## 6.3 Aufgabe

WordPress wurde so konfiguriert, dass es die `schokowpdb` Datenbank verwendet.

Nun wurde noch in der WordPress Konfiguration `/var/www/wp/wp-config.php` der Wert `WP_DEBUG` auf `true` gesetzt, damit Fehler bei der Plugin-Entwicklung einfacher erkannt werden.

Nun wurde in dem Ordner `.../wp-content/plugins` eine PHP Datei erstellt und danach wurden die richtigen Dateiberechtigungen gesetzt.

Damit dieses File als Plugin erkannt und richtig beschrieben wird bedarf es eines Kommentars am Anfang des Files.

```
/*
Plugin Name: SchokoDB Plugin
Author: Daniel M.
Version: 1.0
Description: Ein Plugin fuer die SchokoDB
*/
```

Abbildung 1: Fileheader

Nun wird das Plugin in WordPress erkannt und es kann aktiviert werden.



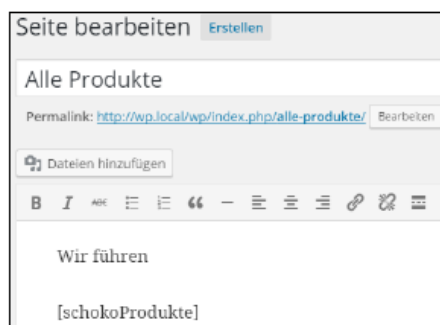
```

1  <?php
2  /*
3   Plugin Name: SchokoDB Plugin
4   Author: Daniel M.
5   Version: 1.0
6   Description: Ein Plugin fuer die SchokoDB
7  */
8
9  function schokoProdukte_func() {
10     global $wpdb;
11     ob_start();
12
13     $res = $wpdb->get_results("select * from Produkt");
14
15     $text = '';
16     foreach($res as $r) {
17         echo $r->bezeichnung;
18
19         <br>
20         <hr>
21     <?php
22         echo '<br>';
23     }
24
25     return ob_get_clean();
26 }
27
28 add_shortcode('schokoProdukte', 'schokoProdukte_func');
```

Abbildung 3: Bisheriges Plugin

Das Plugin, das bis jetzt erstellt wurde, gibt nur alle Produkte der Schokofabrik aus. Dazu wird der Ausgabepuffer gestartet, damit die Ausgabe zuerst zwischengespeichert wird. Danach wird mittels der Datenbankvariable die `get_results` Funktion aufgerufen, welche einen Iterator auf eine Ergebnisliste einer Query zurückliefert. Nun wird mit einer `foreach` Schleife durchiteriert und jede Produktbezeichnung wird ausgegeben. Nach jeder Bezeichnung folgen ein Zeilenumbruch, ein horizontaler Trennstrich und noch ein Zeilenumbruch. Mit der Funktion `ob_get_clean` wird der aktuelle Ausgabepuffer zurückgegeben und danach gelöscht.

Die Seite wird jetzt in WordPress wie folgt erstellt:



```

<?php
/**
 * Plugin Name: SchokoDB-Plugin
 * Author: Kanyildiz Muhammedhizir
 * Version: 3.0
 * Description: Plugin SchokoDB
 */

/**
 * Funktion um alle Schnaepchen auszugeben.
 *
 * Gibt alle Schnaepchen (schaetzwert < 2000 oder preis < 3) der Schokofabrik aus.
 * @return Bezeichnung, Gewicht und Art aller Schnaepchen in einer Tabelle.
 */
function schokoSchnaepchen_func()
{
    $wpschokofabrik = new WPDB('user01', 'user01', 'Schokoladenfabrik', 'localhost');
    ob_start();

    $res = $wpschokofabrik->get_results("SELECT * FROM schnaepchen");

    echo '<table class="table-striped table-hover"><tr><th>Bezeichnung</th><th>Gewicht in g</th><th>ist</th></tr>';
    foreach ($res as $r) {
        ?>
        <tr>
            <td>
                <?php echo $r->bezeichnung; ?>
            </td>
            <td class="small">
                <?php echo $r->gewicht; ?>
            </td>
            <td class="small">
                <?php echo $r->ist; ?>
            </td>
        </tr>
    </td>
    <?php
    }
    echo '</table>';

    return ob_get_clean();
}

/**
 * Funktion um alle Kunstschauen und Details auszugeben.
 *
 * Gibt alle Kunstschauen der Schokofabrik in einem Formular zurueck. Waehlt man eine Kunstschau aus, erhaelt man eine
 * Tabelle aller ausgestellten Kunstwerke inklusive der Platzierung.
 * @return Kunstschauen und Details dazu in einem Formular bzw. in einer Tabelle.
 */
function schokoKunst_func()
{
    $wpschokofabrik = new WPDB('user01', 'user01', 'Schokoladenfabrik', 'localhost');
    ob_start();
    ?>
    Bitte auswaehlen:
    <form method="GET">
        <select name="kunstschau">
            <?php
            $kunstschauen = $wpschokofabrik->get_results("SELECT * FROM kunstview");
            foreach ($kunstschauen as $k) {
                echo '<option value="' . $k->datum . ' ' . $k->name . '">' . $k->datum . ' ' . $k->name . '</option>';
            }
        </select>
        <input type="submit" value="Anzeigen" class="btn btn-default">
    </form>
    <?php

    if (isset($_REQUEST['kunstschau'])) {
        //aufteilen der durch , getrennten Primary Keys
        $requested = explode(',', $_REQUEST['kunstschau']);
        $kunstschaudetails = $wpschokofabrik->get_results($wpschokofabrik->prepare('SELECT bezeichnung, platz FROM platzierung JOIN prodview ON
        platzierung.kunstwerknummer=prodview.nummer WHERE name=%s AND datum=%s', $requested[1], $requested[0]));
        echo '<h4 class="well">Platzierungen f&uuml;r ' . $requested[1] . ' ' . $requested[0] . '</h4>';
        echo '<table class="table-striped table-hover"><tr><th>Bezeichnung</th><th>Platz</th></tr>';
        foreach ($kunstschaudetails as $r) {
            ?>
            <tr>
                <td>
                    <?php echo $r->bezeichnung; ?>
                </td>
                <td class="small">
                    <?php echo $r->platz; ?>
                </td>
            </tr>
        </td>
        <?php
    }
    echo '</table>';
    }
    return ob_get_clean();
}

add_shortcode('schokoSchnaepchen', 'schokoSchnaepchen_func');
add_shortcode('schokoKunst', 'schokoKunst_func');

```

## 6 Aufwandsabschätzung

Arbeit	Stundenabschätzung
Plug-In Erstellen	4:00 h
Installation	0:10 h
Dokumentation	1:00 h

## 7 Literatur Verzeichnisse

Information
<b>GitHub:</b> <a href="https://github.com/mkanyildiz01/TGM-SEW-INSY.git">https://github.com/mkanyildiz01/TGM-SEW-INSY.git</a>