

**Arbeitsprotokoll**

**WordPress PHP Plugin**

**INSY - CMS**

**4CHITT 2015/16**

**Daniel May**

|  |  |
| --- | --- |
|  | **Version 0.1** |
| **Note:** | **Begonnen am 10. März 2016** |
| **Betreuer: Christoph Roschger** | **Beendet am 13. März 2016** |

Inhaltsverzeichnis

[1 Einführung 3](#_Toc445658437)

[1.1 Ziele 3](#_Toc445658438)

[1.2 Aufgabenstellung 3](#_Toc445658439)

[1.3 Hinweise 3](#_Toc445658440)

[2 Ergebnisse 4](#_Toc445658441)

[2.1 Theorie 4](#_Toc445658442)

[2.2 Designüberlegung 4](#_Toc445658443)

[2.3 Allgemeines 4](#_Toc445658444)

[2.4 Datenbank 4](#_Toc445658445)

[2.5 WordPress 5](#_Toc445658446)

[2.6 GitHub Link 7](#_Toc445658447)

[3 Zeitmanagement 7](#_Toc445658448)

[4 Literaturverzeichnis 8](#_Toc445658449)

# Einführung

„Stoesst man bei der Verwendung eines CMS an die Grenzen der verfuegbaren Erweiterungen (Plugins,Extensions,Module,...), bieten alle gaengigen CMS Systeme Schnittstellen zur Anpassung an. Zum Beispiel stellt die Integration externer Datenquellen eine solche Herausforderung dar.“ [ROS]

## Ziele

„Es soll eine Wordpress-Seite aufgesetzt werden, die Daten aus der SchokoDB anzeigen kann, sowohl statisch als auch interaktiv abhaengig von Benutzereingaben.“ [ROS]

## Aufgabenstellung

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

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

1. 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.
2. Auf einer anderen Seite soll der Benutzer die Moeglichkeit haben, eine der in der Datenbank angelegten Kunstschauen auszuwaehlen. Anschlieszend soll zur gewaehlten Kunstschau eine Liste mit allen ausgestellten Kunstwerken samt erreichten Plaetzen angezeigt werden.“ [ROS]

## Hinweise

* „Verwende dazu die mysql-Version der schokoDB (ist im Kurs hochgeladen).
* Nutze Statements um sql injections zu vermeiden.
* Abzugeben sind ein Arbeitsprotokoll sowie das php Plugin

[1] <https://codex.wordpress.org/Writing_a_Plugin>“ [ROS]

# Ergebnisse

## Theorie

In PHP ist es üblich den schließenden Tag wegzulassen, da so automatisch der Schlusstag gesetzt wird und sichergestellt wird, dass keine Leerzeilen eingefügt werden.

## Designüberlegung

Für die zurückzugebenden Listen werden Funktionen erstellt, welche mittels shortcode verwendet werden können. Dafür definiert man eine Funktion und später kann man mittels des Funktionsaufrufs add\_shortcode(kurzname,funktionsname); den Shortcode setzen. In der WordPress Seite lässt sich nun mittels [kurzname] der Rückgabewert der Funktion darstellen.

## Allgemeines

Zuerst wurde WordPress heruntergeladen und ins /var/www Verzeichnis entpackt. Danach wurden die entsprechenden Dateirechte gesetzt. Nun wurde ein ServerAlias /wp gesetzt und ein Allow from All für die Seite gesetzt, damit man sie erreichen kann. Zusätzlich wurde in der Host Datei ein Eintrag für die Adresse mit dem Namen wp.local erstellt.

## Datenbank

Nun wurde in mysql eine Datenbank mit entsprechendem User erstellt.

1. CREATE DATABASE schokofabrik;
2. CREATE USER schokouser@localhost IDENTIFIED by ‘schokouser‘;
3. GRANT ALL PRIVILEGES ON schokofabrik.\* TO   
   schokouser@localhost;
4. FLUSH PRIVILEGES; -- neu laden der privileges

Die Inhalte der Datenbank wurde mittels dem Befehl source und dem beigelegten SQL File eingefügt.

## WordPress

WordPress wurde so konfiguriert, dass es diese 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.

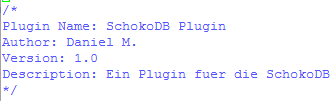


Abbildung 1: Fileheader

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

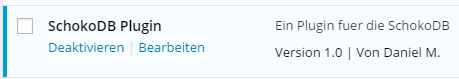


Abbildung 2: aktiviertes Plugin

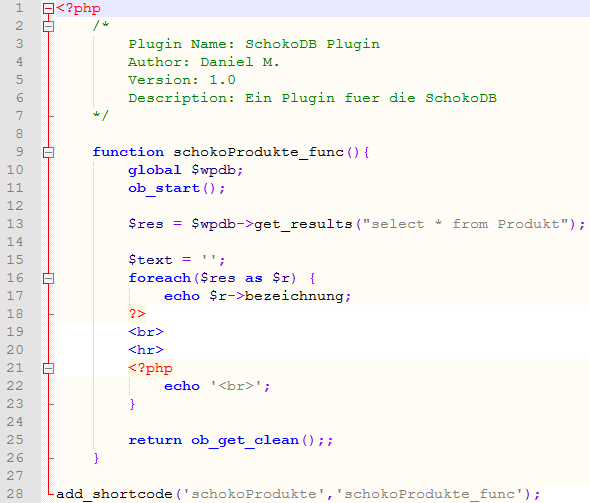


Abbildung 3: Bisheriges Plugin

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:

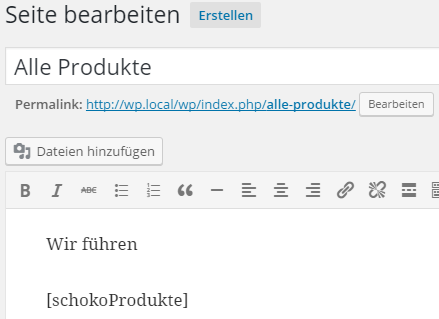


Abbildung 4: Seite für alle Produkte

Nun ist vorerst eine Seite vorhanden die alle Produkte der Schokofabrik ausgibt.

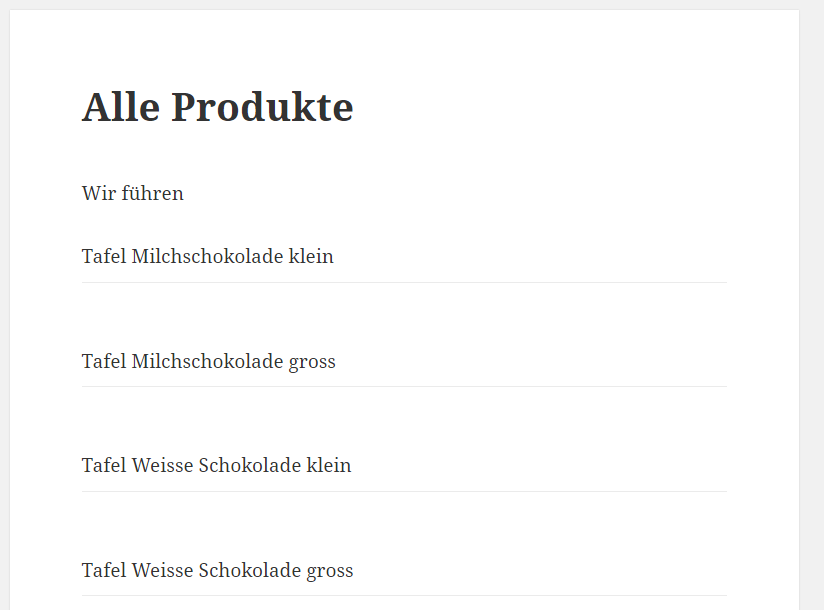


Abbildung 5: Alle Produkte

## GitHub Link

<https://github.com/dmay-tgm/Wordpress-Plugin>

Arbeitsdurchführung (Resultate/Niederlagen)

genauer Testbericht

# Zeitmanagement

Der Aufwand wird wie folgt geschätzt:

|  |  |
| --- | --- |
| **Arbeitsteil** | **geschätzter Aufwand** |
| Übung | 2.5 h |
| Protokoll | 1.5 h |
| **Gesamt** | **4 h** |

Der tatsächliche Aufwand ist wie folgt aufgeschlüsselt:

|  |  |  |
| --- | --- | --- |
| **Durchgeführte Arbeit** | **Datum** | **tatsächlicher Aufwand** |
| Übung | 10.03.2016 | 1 h |
| Protokoll | 13.03.2016 | 1 h |
| **Gesamt** | **10. – 13.03.2016** | **6 h** |

# Literaturverzeichnis

[ROS] Christoph Roschger.   
DATENBANKZUGRIFFE AUS WORDPRESS HERAUS [Online]. Available at: <https://elearning.tgm.ac.at/mod/assign/view.php?id=47516> [abgerufen am 13.03.2016]