

# Projektspezifikation - Webshop

---

## I. Allgemein

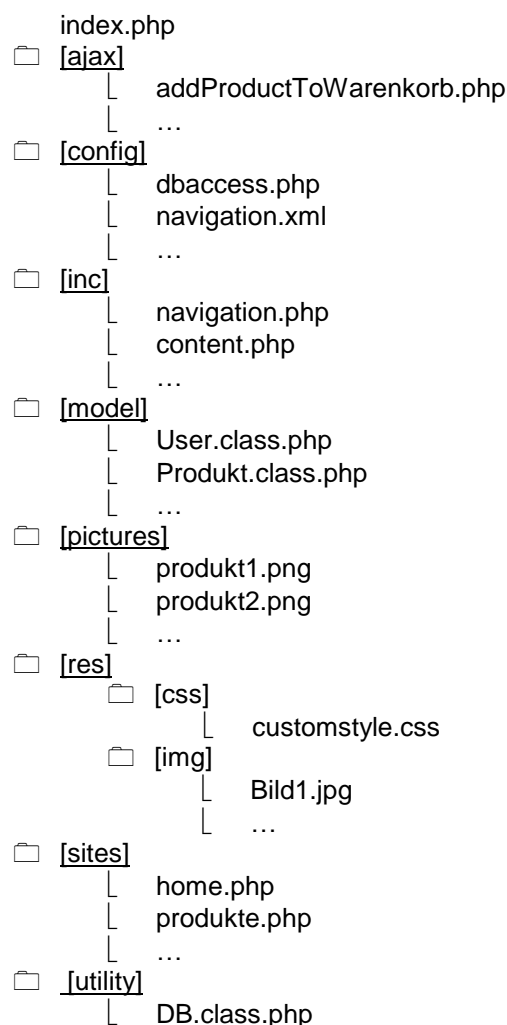
- a. Zum Abschluss der LV Webtechnologien 2 wird ein Projekt umgesetzt:  
BWI- Webshop
- b. Das Projekt kann von 1 bis max. 2 Personen umgesetzt werden.
  - i. Unabhängig von der Gruppenzugehörigkeit
  - ii. Wenn das Projekt von nur 1 Person ausgeführt wird, können die **blau** hervorgehobenen Teile ausgespart werden
- c. Die Angabe lässt Ihnen gewisse Gestaltungs- und Implementierungsfreiheiten. Die Branche des Webshops kann frei gewählt werden (Möbel, Tickets, Elektronik, ...). Folgende **Basis-Features** sollen abgebildet werden:
  - i. Präsentation und Verwaltung von Produkten
  - ii. Warenkorbfunktionalität
  - iii. Kundenverwaltung
  - iv. Bestellungen & Rechnungen
  - v. **Gutscheine verwalten und einlösen**
  - vi. Administrationsbereich
- d. Die Umsetzung des Projekts, wird **sämtliche LV-Inhalte** des Semesters abdecken:
  - i. JavaScript & jQuery (AJAX)
  - ii. PHP & XML
  - iii. PHP & OOP
  - iv. Fileuploads
  - v. Sessions und Cookies
  - vi. MySQLi + Prepared Statements
- e. Achten Sie auf einen **homogenen Prozessablauf** innerhalb der Applikation

## II. Projektstruktur und Grundlayout

- a. Erstellen Sie eine Seite, welches das **Grundlayout** enthält und die Hauptbereiche des Webshops abdeckt sowie etwaige CSS und JS-Libraries einbindet. Die Inhalte aller Bereiche werden mittels include eingebunden. (Header, die Navigation, der Hauptinhaltsbereich ..).

- b. Die **Navigation** wird aus einer **XML-Datei** erzeugt und ist abhängig von den **Berechtigungen** des eingeloggten Users.
- c. Die Projektstruktur des Webshops soll **Komponenten** klar unterteilen.
- d. Die Implementierung des Webshops ist mittels **OOP-Konzept** umzusetzen. Erstellen Sie Klassen für alle Entitäten die im System gebraucht werden (bspw. User, Produkte, ...) und legen Sie eine DB-Service-Klasse an, welche **zentral** die Zugriffe auf die DB implementiert.
- e. Achten Sie beim Implementieren darauf, dass **Sicherheitslücken** möglichst **geschlossen** werden.
- f. Eine **Datenbank**, welche den Webshop abbildet, ist anzulegen.

Beispielstruktur:



### III. Abgabemodalitäten

- a. Das Projekt ist **vollständig** als zip/rar File im Moodle hochzuladen inklusive einer Kopie der MySQL-Datenbank und einem Hinweis auf die Teammitglieder im Dateinamen, bspw:
  - i. **Webshop\_Nachname\_Nachname\_Nachname.zip**
- b. **Kommentieren** Sie komplexere Abläufe/Funktionen im Code
- c. Falls weitere Konfigurationsschritte zur Inbetriebnahme d. Applikation nötig sind, legen Sie eine **readme.txt**-Datei bei.

### IV. Versionen

Version	Datum	Änderungen	Offene Punkte
1.0	13. März 2018	Erstellung Spezifikation	

## Anforderungskatalog

### User-Sicht

*Der Standarduser findet folgende Detail-Funktionalitäten vor:*

- Registrierung/Login
- Produkte ansehen
- Produkte in den Warenkorb legen
- Produkte suchen
- Warenkorb bearbeiten
- Produkte bestellen (kein Bezahlvorgang!)
- Bestellungen und Rechnungen einsehen
- Eigene Daten bearbeiten

### 1) Users-Registrierung

- a. Jeder User kann ein neues Benutzerkonto für den Webshop anlegen
- b. Beim Erfassen der Users sind folgende Daten anzugeben:
  - i. Anrede
  - ii. Vorname
  - iii. Nachname
  - iv. Adresse
  - v. PLZ
  - vi. Ort
  - vii. Emailadresse
  - viii. Benutzername
  - ix. Passwort
  - x. Zahlungsinformationen
- c. Überprüfen Sie alle Parameter sowohl mittels HTML5/JavaScript-Validierung also auch serverseitig auf Vollständigkeit und Richtigkeit. Ergänzen Sie etwaige HTML5-Unvollständigkeiten. Das Passwort muss immer 2x angegeben werden und die Werte werden auf Übereinstimmung geprüft. Erst wenn alle Daten validiert wurden, wird der neue User in der DB angelegt. Das Passwort wird verschlüsselt in der DB abgelegt.
- d. Legen Sie manuell in der Datenbank einen weiteren User an, der als Administrator gekennzeichnet wird. Die Rechte des Administrators unterscheiden sich von herkömmlichen Usern.

- e. Wenn der User angelegt wurde, kann die Login-Funktion genutzt werden:

## 2) User-Login

- a. Erstellen Sie ein Login-Formular welches Usernamen und Passwort entgegennimmt
- b. Weiters enthält das Formular eine Checkbox mit der Option „Login merken“
- c. Überprüfen Sie die Werte gegen die Datenbank. Nur wenn es einen User in der DB mit den Daten, bekommt der/die UserIn die Rückmeldung „Hallo User“. Der Login-Status ist permanent auf der Seite sichtbar. Eingeloggte User sehen einen Logout-Button.
- d. Auf Basis des eingeloggten Users, werden unterschiedliche Menüs generiert.
  - i. Ist kein User eingeloggt, sind nur die Bereiche „Home“, „Produkte“ und „Warenkorb“ zu sehen
  - ii. Ein eingeloggter User sieht die Einträge „Home“, „Produkte“, „Mein Konto“, „Warenkorb“
  - iii. Ein Administrator sieht „Home“, „Produkte bearbeiten“, „Kunden bearbeiten“ und „**Gutscheine verwalten**“
- e. Falls der User „Login merken“ aktiviert hat, wird auch ein Cookie angelegt, welches den User automatisch einloggt, wenn es gesetzt ist.
- f. Achten Sie darauf, dass beim Schließen des Browsers auch die Session gelöscht wird und der User ausgeloggt ist, es sei denn, es wurde ein Cookie gesetzt, um sich den Login zu merken, dann wird der User beim erneuten Öffnen der Web-Adresse als eingeloggt angezeigt.

The image shows a mockup of a web application. On the left is a 'Login' form with the following elements: a title 'Login', a 'Username' label with an input field containing 'admin', a 'Passwort' label with a password input field (masked with dots), a checkbox labeled 'angemeldet bleiben', and a 'Submit' button. Overlaid on the right is a white rectangular box with a shadow. It contains the text 'Willkommen!' in a large, bold, blue font, followed by 'Sie sind eingeloggt ...' in a smaller blue font, and a 'Logout!' link in blue at the bottom.

### 3) Produkte ansehen

- a. Im Bereich „Produkte“ werden Produkte angezeigt sowie Kategorien. Die erste Kategorie wird standardmäßig ausgewählt und es werden die aktuellen Produkte dieser Kategorie angezeigt. Wechselt der User die Kategorie, werden neue Produkte geladen. Jedes Produkt wird angezeigt mit Name, Bild, Preis und Bewertung.
- b. Bei jedem Produkt gibt es die Option via Link „In den Warenkorb legen“. Klickt der User auf den Link, wird das gewählte Produkt im Warenkorb abgelegt.
- c. Das Ablegen von Produkten im Warenkorb soll via AJAX erfolgen und ohne Page-Reload funktionieren.
- d. Neben einem kleinen Warenkorbsymbol auf der Seite wird die aktuelle Anzahl der sich gerade im Warenkorb befindlichen Produkte angezeigt – diese Zahl wird ebenfalls via AJAX aktualisiert. Beim Stöbern durch die Produkte und beim „Einkaufen“ soll der User nie die Seite verlassen müssen.
- e. [Produkte können auch via Drag’n’Drop auf das Warenkorb-Symbol gezogen werden, um darin abgelegt zu werden.](#)
- f. Navigiert der User in den Bereich „Warenkorb“ werden alle Produkte als Liste angezeigt, welche im Warenkorb abgelegt wurden sowie der Gesamtpreis aller Produkte. Jedes Produkt kann wieder aus dem Warenkorb entfernt oder mehrfach abgelegt (Anzahl erhöhen) werden.

### 4) Produktsuche (mit Continuous Search Filter)

- a. Fügen Sie der Produktansicht ein Suchfeld hinzu. Tippt der User einen Buchstaben ein, beginnt bereits die Suche:
- b. Basierend auf der Eingabe werden Produkte aus der Datenbank geholt und die Ergebnisse angezeigt, ohne einen Reload der Seite.

### 5) Produkte bestellen

- a. Im Bereich Warenkorb kann der User, sofern Produkte im Warenkorb sind, die Produkte bestellen.
- b. Bestellungen sind nur möglich, wenn der User angemeldet ist. Nach dem Login, sollen immer noch alle Daten im Warenkorb sichtbar sein.

- c. Bei der Bestellung muss der Kunde eine seiner Zahlungsmöglichkeiten auswählen oder er bezahlt mittels Gutschein. Der Gutscheinwert wird vom Gesamtbetrag abgezogen bzw. bleibt der Restwert bei nur teilweiser Einlösung erhalten.
- d. Sind Daten unvollständig, kann die Bestellung nicht abgeschlossen werden und die Produkte verbleiben im Warenkorb.
- e. Die Bestellung wird in der DB abgelegt, sodass jederzeit nachvollziehbar ist, welcher Kunde welche Produkte bestellt hat. Ein Kunde solle jederzeit die Details zur Bestellung einsehen können.

## 6) Kundenkonto einsehen

- a. Im Bereich „Mein Konto“ kann der User seine Daten einsehen und bearbeiten (Stammdaten bearbeiten, neue Zahlungsmöglichkeiten hinzufügen).
- b. Achten Sie drauf, dass sensible Informationen nicht vollständig angezeigt werden und beim Ändern von Daten, das Passwort verlangt wird.
- c. Sämtliche Bestellungen des Users werden nach Datum aufwärts sortiert angezeigt. Zu jeder Bestellung können die Details eingesehen werden. Der User kann zu Jeder Bestellung eine Rechnung drucken. In diesem Fall wird eine Rechnungsnummer generiert und die Rechnung wird mit Positionen, Datum und Anschrift in einem neuen Fenster angezeigt.

## **Administrator-Sicht**

*Der Webshop-Admin findet folgende Detail-Funktionalitäten vor:*

- Produkte anlegen
- Produkte bearbeiten
- Kunden verwalten
- Gutscheine verwalten

### **1) Produktverwaltung**

- a. Im Bereich „Produkte bearbeiten“ kann der Administrator sowohl neue Produkte anlegen (zumindest mit Name, Beschreibung, Bewertung, Preis, Foto) als auch Produkte bearbeiten und löschen.
- b. Für jedes Produkt kann ein Produktfoto hochgeladen werden.

### **2) Kunden verwalten**

- a. Der Admin hat die Möglichkeit, alle existierenden Kunden aufzulisten und deren Bestelldetails einzusehen. Jeder Kunde kann vom Admin deaktiviert werden, somit ist kein Login mehr bzw. keine Einkäufe mit diesem Account möglich.
- b. Einzelne Produkte können aus der Bestellung des Kunden entfernt werden und sind auch für den Kunden nicht mehr sichtbar.

### **3) Gutscheine verwalten**

- a. Weiters gibt es für den Admin die Option, Gutscheincodes zu erstellen. Jeder Gutscheincode wird zufällig generiert und besteht aus einem 5-stelligen alphanumerischen Code. Je Gutschein werden Wert sowie ein Gültigkeitsdatum angelegt.
- b. Der Admin hat eine Übersicht über alle Gutscheincodes, deren Datum sowie Wert. Bereits eingelöste bzw. abgelaufene Gutscheine werden ebenfalls gelistet, jedoch entsprechend markiert.