

Duale Hochschule Baden-Württemberg

Mannheim

**Dokumentation**

**GC DHBW**

Wirtschaftsinformatik – Sales & Consulting

Verfasser: Denis Maag

Steffen Schmitz

Kurs: WWI 13 SC A

Studiengangsleiter: Prof. Dr. Frank Koslowski

Betreuer: Prof. Dr. Konrad Preiser

# Anforderungen

Im Rahmen des Moduls Verteile Systeme soll eine Vereinswebseite entwickelt werden, die verschiedene Standards der Webprogrammierung abdeckt. Folgende Anforderungen wurden gestellt:

|  |  |
| --- | --- |
| **Anforderung** | **Status** |
| HTML5 und CSS3 als Grundlage für die einzelnen Webseiten | Erledigt |
| Mind. 1 PHP-Seite mit mind. 1 sinnvollen PHP-Funktionalität | Erledigt |
| Mind. 1 DB-Zugriff auf MySQL von PHP aus | Erledigt |
| Mind. 1 JSP-Seite mit einer sinnvollen Funktionalität | Erledigt |
| Mind. 1 Servlet mit einer sinnvollen Funktionalität | Erledigt |
| Mind. 1 JavaBean | Erledigt |
| Mind. 1 DB-Zugriff mit JDBC | Erledigt |
| Mind. 1 server-seitig erstellte, dynamische Grafik | Erledigt |
| Mind. 1 client-seitig erstellte oder bearbeitete Grafik | Erledigt |
| Optional: SVG-Grafik | Erledigt |
| An- und Abmeldung eines Benutzers mit einem gewissen Mehrwert für diesen | Erledigt |
| Mind. 1 HTML-Formular mit einer sinnvollen Auswertung der Eingaben | Erledigt |
| Mind. 1 Einsatz eines Cookies, das selbst gesetzt und gelesen wird | Erledigt |
| Optional: Ein Einsatzbeispiel für Local-Storage | Erledigt |
| Mind. 1 sinnvolle JavaScript-Funktionalität | Erledigt |
| Mind. 1 Beispiel für AJAX | Erledigt |
| Optional: Ein Beispiel für die Vorzüge von WebSockets |  |
| Mind. 1 selbst programmiertes Java-Applet |  |
| Unterstützung der Anfahrt mit Google Maps | Erledigt |
| Social-Anbindung | Erledigt |
| Exemplarisches Anpassen der Web-Anwendung an die Bedürfnisse eines mobilen Endgerätes | Erledigt |

# 2 Umsetzung

Im Folgenden Kapitel werden die Anforderungen aufgeführt und anschließend gezeigt, wie diese im Rahmen der Webseite des GC DHBW umgesetzt wurden.

## 2.1 HTML 5 und CSS3 als Grundlage für die einzelnen Webseiten

HTML (Hypertext Markup Language) dient zur Strukturierung von Inhalten, wie Text und Bildern, im Internet. Die Formatierung des Textes erledigt das CSS (Cascading Style Sheet). Im folgenden Beispiel ist der HTML Code der Startseite auszugsweise abgebildet. Die sichtbaren Kästen werden durch sogenannte „Articles“ konstruiert, die mit verschiedenen „Divs“ beispielsweise Überschrift und Text voneinander abgrenzen. Jedes Element wird durch einen Tag (<>) geöffnet und einen Tag mit einem Slash (</>) wieder geschlossen. So entsteht eine Baumstruktur, die vom Browser ausgewertet wird.



Abb.1: HTML Code

Wie in der Grafik zu sehen ist hat der Article die Klasse „post“ und die Überschrift h1 die Klasse „post-title“. Klassen werden im CSS durch einen Punkt (.) angezeigt, während IDs mit einem Hashtag (#) verdeutlicht werden. Im CSS Auszug unten ist sehr gut zu sehen, dass die Überschrift eine Schriftgröße von 28px erhält und groß geschrieben wird (font-size: 28px und text-transform: capitalize).

Unsere Webseite baut vollständig auf einer Kombination aus HTML und CSS auf und erhält somit ein einheitliches Erscheinungsbild, da gleiche Elemente dank des CSS gleich dargestellt werden. In Abbildung 3 ist das Bild des Anwenders dargestellt.

## 

Abb.2: CSS Code



Abb.3: Startseite

## 2.2 Mind. 1 PHP-Seite mit mind. 1 sinnvollen PHP-Funktionalität

PHP dient zur Erstellung von dynamischen Webseiten und Web-Anwendungen. In unserem Projekt nutzen wir PHP um die Inhalte der Unterseiten in unsere start.php dynamisch zu laden, damit das Hintergrundbild und ähnliches nicht stets neu heruntergeladen werden müssen. Hierzu wird dem PHP Code über den Browser Link ein Schlagwort übergeben, dass er einer weiteren Datei zuordnet und den Inhalt dieser lädt. Dies ist in Abbildung 4 zu sehen.



Abb.4: PHP Skript

PHP kann dabei in HTML Code eingebunden werden und wird durch <?php ?> geöffnet und geschlossen. Außerdem kann PHP HTML Code zurückgeben, wie im obigen Beispiel. Die einzubindenden Dateien haben den Dateityp .php, geben aber nur HTML Code zurück.

## 2.3 Mind. 1 DB-Zugriff auf MySQL von PHP aus

PHP kann außerdem zur Abfrage von Datenbanken dienen. Die Datenbank im GC DHBW Projekt wurde über die Weboberfläche angelegt und befüllt und kann bei der Anmeldung ausgelesen werden. Der Anwender übergibt einen Anwendernamen und das dazugehörige Passwort und wenn diese mit den Daten der Datenbank übereinstimmen wird ein Cookie (Dazu später mehr) gesetzt, dass den Zugriff auf die Links des „Mitglieder Only“ Menüpunkts freigibt.

Die „login.php“ zeigt hierzu ein HTML Formular (Dazu später mehr) an, dass die Eingaben entgegen nimmt und an die „anmeldung.php“ übergibt. Diese ruft „settings.php“ auf, die eine Verbindung zur Datenbank erstellt. „anmeldung.php“ prüft im Anschluss die Benutzereingabe und leitet mit gesetztem Cookie zurück auf die Startseite.

Hierzu liefert PHP native Methoden um mit MySQL Datenbanken zu arbeiten.

## 2.4 Mind. 1 JSP-Seite mit einer sinnvollen Funktionalität

Die JSP Seite des GCDHBW Projektes wird nur angemeldeten Mitgliedern angezeigt. Sie gibt mit Hilfe eines JavaBean (Dazu später mehr) das aktuelle Datum zurück. JSP Seiten sind Java Server Pages und ein Konkurrent von PHP bei der Erstellung dynamischer Webseiten. Mit Hilfe eines <% %> Tags kann Java verarbeitet werden. Alternativ hierzu können mit einem jsp:useBean Tag JavaBeans importiert und ausgeführt werden.



Abb.5: JavaBean-Aufruf aus einer JSP

## 2.5 Mind. 1 Servlet mit einer sinnvollen Funktionalität

Ein Servlet ist eine Java-Klasse, die Anfragen eines Clients entgegennimmt und beantwortet. Das Servlet des GC DHBW gibt mit Hilfe einer Datenbankabfrage über JDBC (Dazu später mehr) die aktuellen Veranstaltungen aus der „veranstaltungen“ Tabelle zurück. Hierbei wird der Inhalt beim Aufruf des Servlets erstellt und anschließend im Browser angezeigt wird.



Abb.6: Ausgabe des Veranstaltungsservlets

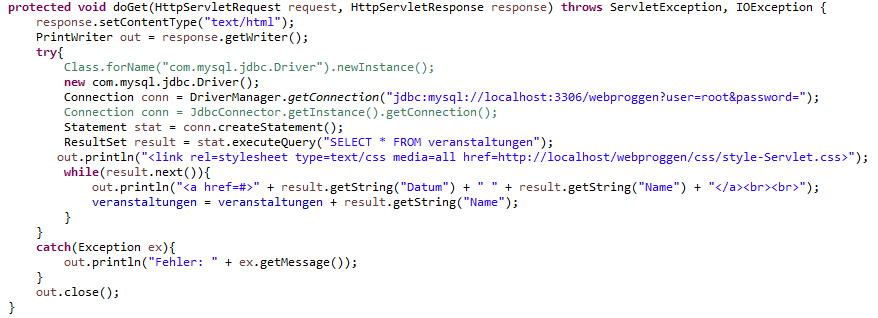


Abb.7: Implementierung der JDBC Datenbankabfrage

Hierzu wird die doGet Methode der Super-Klasse HttpServlet überschrieben, die auf einen HTTP GET Aufruf etwas zurückgibt. out.println(„*text*“); gibt HTML Code zurück, der vom Client angezeigt wird. In der while-Schleife werden alle Resultate der Datenbankabfrage ausgegeben.

## 2.6 Mind. 1 JavaBean

JavaBeans dienen häufig als Container zur Datenübertragung. Sie bieten öffentliche Getter und Setter, mit denen auf die Variablen zugegriffen werden kann. Die JavaBean des GC DHBW stellt einen Kalender bereit, der den Wochentag, das Jahr oder den aktuellen Monat zurückgeben kann. Die Mitglieder des GC DHBW erhalten durch Aufruf der dazugehörigen JSP das vollständige Datum des aktuellen Tages ausgegeben, was eine Kombination der Attribute des Kalenders darstellt.

Das JavaBean implementiert hierzu einen Kalender und Getter Methoden. Beim Aufruf wird die Property „date“ abgerufen, die den vollständigen Eintrag des aktuellen Tages zurückgibt.



Abb.8: Getter für das aktuelle Datum

## 2.7 Mind. 1 DB-Zugriff mit JDBC

JDBC (Java Database Connectivity) dient zur Verbindungen von Java zu verschiedenen Datenbanken. Im Rahmen des Projektes GC DHBW wird aus einem Servlet via JDBC auf eine MySQL Datenbank zugegriffen und die Ergebnisse der Anfrage im Anschluss ausgegeben.

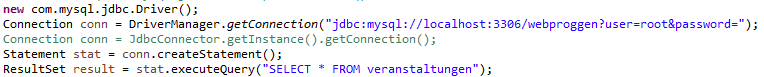


Abb.9: JDBC Query

Wie in Abbildung 9 zu sehen ist wird eine JDBC Verbindung hergestellt, die direkt auf die anzufragende Datenbank verweist. Im Anschluss wird ein SQL Statement erstellt, das die Tabelle ausliest und als Ergebnis zurückgibt.

## 2.8 Mind. 1 server-seitig erstellte, dynamische Grafik

Die server-seitig erstellte, dynamische Grafik wurde im GC DHBW Projekt im Rahmen eines Captchas umgesetzt. Die legt über einen Hintergrund und ein bestimmtes Muster einen zufällig generierten Text und gibt diesen als Bild zurück. Der Anwender muss den Text dieses Bildes dann eingeben, um sich zu verifizieren.

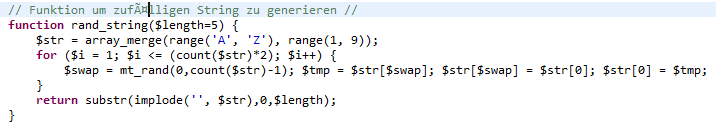


Abb.10: Generieren eines zufälligen Strings

Der Client erhält dabei nur eine Bilddatei, die der Server vorher erstellt hat und damit unterscheidet sich das Captcha von der client-seitig erstellten Grafik

## 2.9 Mind. 1 client-seitig erstellte oder bearbeitete Grafik

Die client-seitig erstellte Grafik ist eine so genannte Canvasgrafig. Dies steht für „Leinwand“ und bietet eine Fläche, die „bemalt“ werden kann. Im Rahmen des GC DHBW Projektes wird die Navigationsleiste durch eine Uhr ergänzt, die die jeweils aktuelle Zeit via JavaScript erfragt und ausgibt.



Abb.11: Canvas-Grafik

Hierzu wurde ein <canvas></canvas> Block angelegt, in dem die Funktion „drawClock()“ aufgerufen wird.

## 2.10 An- und Abmeldung eines Benutzers mit einem gewissen Mehrwert für diesen

Die Anmeldung eines Anwenders erfolgt wie im Kapitel 2.3 beschrieben mittels einer MySQL Datenbankabfrage mit Hilfe von PHP. Das dabei gesetzte Cookie wird automatisch gelöscht, sobald der Anwender den Browser schließt und damit wird der Anwender automatisch abgemeldet.

Als angemeldetes Mitglied des GC DHBW erhält man über die Menüleiste Zugriff auf unsere Fun Fact Seite, auf das Caddy-Tool und kann das aktuelle Datum abfragen.

## 2.11 Mind. 1 HTML-Formular mit einer sinnvollen Auswertung der Eingaben

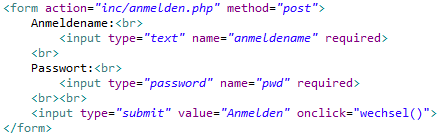
Die Abfrage der Anmeldedaten eines Anwenders erfolgt über ein HTML Formular, das den Namen und das Passwort des Mitglieds an die „anmelden.php“ übergibt. Dabei setzt sich das Formular aus zwei Feldern zur Texteingabe zusammen und einem Knopf, über den man die Anmeldung absenden kann. 

Abb.12: HTML Formular

Dem HTML Formular können in der neuesten Version außerdem verschiedene Typen übergeben werden. In Abbildung 12 sieht man, dass das Passwort den Typ „password“ hat, was bedeutet, dass der eingegebene Text nicht im Klartext angezeigt wird. Die bietet zudem verschiedene Kontrollmöglichkeiten, beispielsweise bei E-Mails oder erleichtert bei mobilen Geräten die Datumseingabe.

## 2.12 Mind. 1 Einsatz eines Cookies, das selbst gesetzt und gelesen wird

Sobald ein Mitglied angemeldet ist, wird ein Cookie gesetzt, dass dessen Vornamen beinhaltet. In der Navigationsleiste wird mittels PHP geprüft, ob das Cookie gesetzt und bei positiver Rückmeldung der Menüpunkt speziell für Mitglieder eingeblendet.

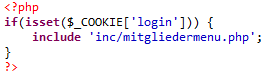


Abb.13: Cookie auslesen

## 2.13 Mind. 1 sinnvolle JavaScript Funktionalität

JavaScript wird im Rahmen der Ajax Funktion implementiert. Siehe hierzu Kapitel 2.14.

## 2.14 Mind. 1 Beispiel für Ajax

## 2.15 Mind. 1 selbst programmiertes Java-Applet

## 2.16 Unterstützung der Anfahrt mit Google-Maps

## 2.17 Social Anbindung

## 2.18 Exemplarisches Anpassen der Web-Anwendung an die Bedürfnisse eines mobilen Endgerätes

# 3 Fazit