

## **WPF Moderne Webanwendungen**

WS17/18

Dozent: Andre Kasper

*Matthias Stolz 11103572*

*Jürgen Stolz 11107534*

*David Jovanoski 11117136*

*Paulo Vinicius Duarte Penha Ribeiro Pinto 11104945*

[**WPF Moderne Webanwendungen**](#_gbqsdhyvq1lh) **0**

[Einleitung](#_pxz0dcdurhfu) 2

[Umsetzung](#_s0ffygx05avl) 2

[CSS](#_silqromodg1x) 2

[Responsive Design](#_ery648af8x95) 2

[Session (Login User/Admin)](#_choia8ye4nrw) 2

[Datenbankanbindung (MySQL)](#_1dtw03b04ha1) 2

[Ajax Request mit Antwort im JSON-Format](#_mvxtov2n7drc) 2

[JS-Funktionen](#_hn5v2f8lcjqv) 3

[JS-Framework/Toolkit/Node.js mindestens einmal genutzt](#_wljbrx4x8i5e) 3

[Minimierte CSS-/JS-Datei](#_aierdrz88gda) 3

[Komprimierung mit Deﬂate oder GZIP via htaccess oder vorkomprimiert](#_nrns8cmitdtd) 3

[Caching mit htaccess oder PHP (z.B. phpfastcache)](#_yzu9itp2yge8) 3

[Suchmaschinenoptimierungen](#_9x38xqc3dm) 3

[Fazit](#_pqydzjkrdtr4) 3

### Einleitung

Das von uns realisierte Projekt ist eine einfache Fußball Seite mit einem Steckbrief, indem sich Nutzer anmelden müssen um den Inhalt zu sehen. Das Projekt wurde mit einer MySQL Datenbank implementiert.

Die Login Daten für unsere Login Funktion lauten:

Benutzer: Max

Passwort: 123

### Umsetzung

Alle Anforderungen an die Webseite wurden erfolgreich implementiert.

#### CSS

Die CSS Datei “style.css” und ‘‘loginStyle.css“ liegen in einem separaten Ordner “public” im Projektordner. Es wurde kein Inline-CSS verwendet.

#### 

#### Responsive Design

In der ‘‘style.css“ und ‘‘loginStyle.css“ haben wir hauptsächlich als Größenangaben Prozentwerte genutzt. Außerdem wird die Breite auf 100% der Bildschirmbreite gesetzt, falls jemand etwas am Browser ändert wird auch das Bild geändert.

#### Session (Login User/Admin)

Sobald der Nutzer seine Daten gesendet hat, wird in der Datenbank überprüft, ob das abgesendete Passwort zu dem Nutzer übereinstimmt. Ist dies der Fall werden der Benutzername und das Passwort, als Session-Variablen gespeichert(server.js). Nur ein Angemeldeter Nutzer mit gesetzter Session-Variable kann den Inhalt unserer Seite sehen.

#### Datenbankanbindung (MySQL)

Die Datenbankabfragen wurden in der server.js gespeichert und mit hilfe des Node Modul „mysql“ umgesetzt. Die Logindaten der Datenbank werden in db.sql erstellt.

#### Ajax Request mit Antwort im JSON-Format

Über den Button „erzähl mir einen Witz“ wird ein Ajax request über die Funktion loadDoc() (main.js) ausgeführt. Als Antwort erhält man einen Witz im JSON Format.

#### JS-Funktionen

Als JavaScript Funktionen haben wir eine simple Methode zur Addition geschrieben.

Die Funktion Calculate() (main.js) liest zwei Werte ein und wirft das Ergebnis als alert aus.

#### JS-Framework/Toolkit/Node.js mindestens einmal genutzt

Zur umsetzung des Projektes wurde Node.js genutzt

#### Minimierte CSS-/JS-Datei

Die CSS- und JS-Dateien wurden alle auf Mindestgröße minimiert (Alle überflüssigen Zeichen entfernt). Jede minimierte Datei ist mit Compressed vorgesehen.

#### Komprimierung mit Deﬂate oder GZIP via htaccess oder vorkomprimiert

Das Komprimieren erfolgt bei uns mit dem Modul mit Deflate und GZIP. Die Angaben an den Server, wie es komprimiert werden soll, stehen in der “.htacces” Datei.

#### Caching mit htaccess oder PHP (z.B. phpfastcache)

Angaben an den Server, wie komprimiert werden soll, stehen in der “.htacces” Datei.

#### Suchmaschinenoptimierungen

Die Optimierungsmaßnahmen:

-Webseite Titel würde hinzugefügt

-Passende Keywords/Tags hinzugefügt

-Dead-Link wurde entfernt

-Iframe wurde auch entfernt

-Responsive Webdesign

### Fazit

Bei der Realisierung der Webseite haben wir einige neue Technologien kennengelernt, unter anderem das Caching von Webseiten. Außerdem haben wir im Umgang mit NodeJS einiges ausprobiert und so viel neues kennen gelernt.