

## UE1 – HTML5, CSS3, WAI und JavaScript (30 Punkte)

Ziel dieses Übungsbeispiels ist die Erstellung von statischen, validen und barrierefreien Webseiten, sowie einfacher Scripts für client-seitige Funktionalität. Serverseitige Funktionalität muss in dieser Übung nicht berücksichtigt werden.

Deadline der Abgabe via TUWEL<sup>1</sup>: **Sonntag, 3. April 2016 23:55 Uhr**

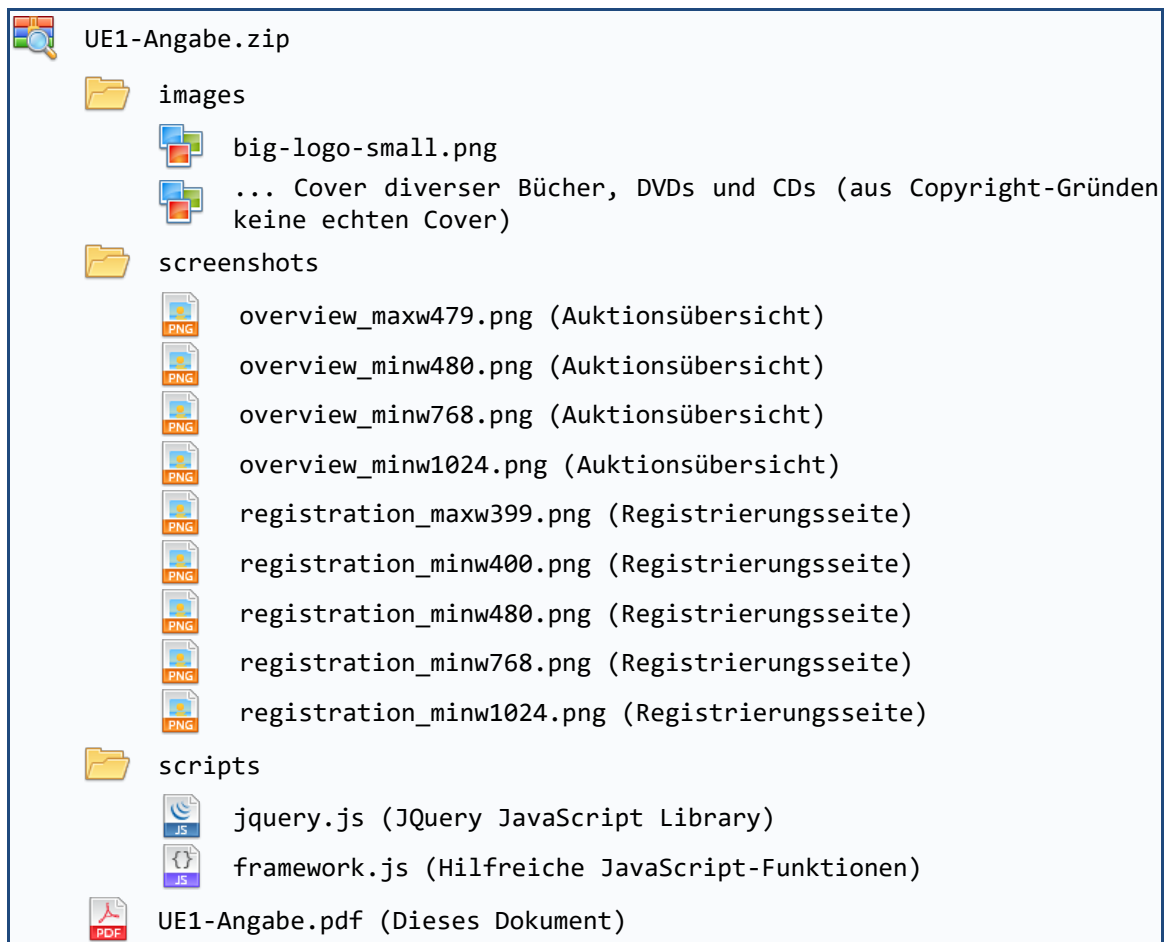
**Nur ein Gruppenmitglied muss die Lösung auf TUWEL abgeben.**

### BIG Bid

BIG BID ist eine Online-Plattform, auf der registrierte Benutzerinnen und Benutzer Bücher, CDs und DVDs ersteigern können. Um sicherzustellen, dass die Benutzerinnen und Benutzer nur Gebote abgeben, die sie sich auch tatsächlich leisten können, müssen sie vor der Teilnahme an Auktionen Guthaben bei BIG Bid kaufen. Neue Gebote werden dann von diesem Kontostand abgebogen. Wenn eine Benutzerin bzw. ein Benutzer überboten wird, dann wird ihr oder ihm der Betrag wieder gutgeschrieben. Nur volljährige Personen können sich bei BIG Bid registrieren.

### Angabe

Diese Angabe umfasst folgende Dateien:



<sup>1</sup> <https://tuwel.tuwien.ac.at/mod/assign/view.php?id=246104>

In der Angabe finden Sie Screenshots eines Registrierungsformulars (screenshots/registration\_\*.png) und einer Produktübersichtseite (screenshots/overview\_\*.png). Ihre Aufgabe ist es, diese beiden Seiten als responsive Webseiten mit HTML5, CSS3 und JavaScript umzusetzen.

Auf der Registrierungsseite können sich neue Benutzerinnen und Benutzer bei BIG Bid registrieren. Hierzu geben sie ihre Kontaktdaten an und wählen ein Passwort. Die Lieferadresse ist optional. Sie kann auch zu einem späteren Zeitpunkt bekannt gegeben werden (nicht Teil der Übung). Wählen Sie für die Formularfelder passende Typen entsprechend der erwartenden Eingabewerten.

Die Produktübersicht zeigt sowohl laufende als auch bereits abgelaufene Auktionen an. Letztere werden etwas blasser dargestellt. Bei jedem Produkt wird Produktbild, Produktname, aktuelles Höchstgebot, verbleibende Bietzeit und Höchstbietender bzw. Höchstbietende angezeigt. Jene Produkte, für welche die angemeldete Benutzerin oder der angemeldete Benutzer ein Gebot abgegeben hat, werden orange hervorgehoben. Die Sidebar bietet den Benutzerinnen und Benutzern eine Übersicht über ihren aktuellen Kontostand, Auktionen, an denen sie sich beteiligt haben, und zuletzt angesehene Produkte.

Auf beiden Seiten befindet sich oben links ein Header mit dem entsprechenden Logo und dem Namen der Anwendung („BIG Bid“) und oben rechts die Navigation mit dem Link zum Anmelden bzw. Abmelden. Das Layout beider Seiten soll eine passende Mindest- und Maximalbreite haben und sich wie in den Screenshots angezeigt für verschiedene Bildschirmgrößen anpassen.

Implementieren Sie die beiden Seiten mit Hilfe von **HTML5** und **CSS3** unter der Befolgung des **WAI Conformance Level Double-A** (oder höher). Beachten Sie daher, dass Sie die richtigen, semantischen Elemente in Ihrem HTML5-Code auswählen. Achten Sie außerdem darauf, dass die Darstellung der Inhalte **ohne** Stylesheets (z.B. für Screen-Reader) im Sinne der WAI ebenso gewährleistet ist, um eine komfortable Verwendung für alle Benutzerinnen und Benutzer zu ermöglichen.

Das Aussehen aller Seiten ist so exakt wie möglich einzuhalten. Insbesondere gilt das auch für Texte, die in deutscher Sprache ausgewiesen sind. Verwenden Sie ausschließlich die Bilder, die in den Angaberessourcen (Ordner images) zur Verfügung gestellt wurden. Achten Sie auch vor allem darauf, dass die Ausgabe in den gängigsten Browsern (zumindest Mozilla Firefox 44, Microsoft Edge und Google Chrome 48) keine Unstimmigkeiten oder Probleme verursacht.

## JavaScript

Entwickeln Sie darüber hinaus folgende Funktionalität mit JavaScript:

Die folgenden Felder des Registrierungsformulars sollen mittels JavaScript während der Eingabe validiert werden. Tritt ein Validierungsfehler auf, so ist eine entsprechende, aussagekräftige Meldung bei dem entsprechenden Feld anzuzeigen. Die Fehlermeldungen sollen nicht sofort beim Laden der Seite angezeigt werden, sondern erst nachdem eine ungültige Eingabe erfolgt ist.

- **Anrede:** Pflichtfeld.
- **Vorname:** Pflichtfeld, aber sonst keine Einschränkungen.
- **Nachname:** Pflichtfeld, aber sonst keine Einschränkungen.
- **Geburtsdag:** Pflichtfeld. Überprüfen Sie, ob es sich bei der Eingabe um ein gültiges Geburtsdatum handelt, welches mindestens 18 Jahre in der Vergangenheit liegt. Um Sie bei der Überprüfung zu unterstützen, stellt `framework.js` die Funktion `getNormalizedDateString` zur Verfügung. Dieser

Funktion können Sie den CSS-Selektor auf das entsprechende Datumsfeld (bspw. #birthdate) übergeben und erhalten ein Datum im Format „dd.mm.yyyy“ zurück.

- Email: Pflichtfeld. Das Format muss der Regular Expression `/^\S+@\S+\.\S+$/` entsprechen, um die Benutzerinnen bzw. Benutzer auf grobe Tippfehler hinzuweisen.
- Passwort: Pflichtfeld. Mindestlänge von 4 Zeichen und Maximallänge von 8 Zeichen.
- Straße/Hausnummer: Optional. Keine Einschränkungen
- PLZ/Ort: Optional. Keine Einschränkungen
- Land: Optional. Keine Einschränkungen
- AGB: Die Checkbox muss aktiviert sein

Achten Sie darauf, dass der Benutzer bzw. die Benutzerin die Daten nur dann an den Server übermitteln kann, wenn im Formular keinerlei Validierungsfehler enthalten sind. Das bedeutet, der „Registrieren“-Button darf nur aktiviert sein (Aussehen wie „Abmelden“-Button), wenn alle Pflichtfelder, d.h. Anrede, Vorname, Nachname, Geburtstag, Email und Passwort, ausgefüllt sind und das Formular keine Validierungsfehler enthält. Ansonsten ist der Button inaktiv und kann vom Benutzer bzw. Benutzerin nicht gedrückt werden (graue Darstellung, siehe Screenshot).

Verwenden Sie zur Validierung einerseits die Validierungsmöglichkeiten von HTML5, andererseits eine eigene JavaScript-Implementierung als Fallback. Um zu überprüfen, ob HTML5-Formularvalidierung möglich ist, können Sie die Funktion `hasFormValidation` aus `framework.js` verwenden. Bei der Überprüfung des Geburtsdatums müssen Sie zusätzlich zu `hasFormValidation` noch `hasNativeDateInput` aufrufen, da es Browser gibt, die zwar HTML5-Formularvalidierung ermöglichen, jedoch das Datumsfeld nicht unterstützen (zB Mozilla Firefox).

In der ersten Übung ist noch keine serverseitige Funktionalität und Anwendungslogik zu implementieren, insbesondere müssen Daten nicht gespeichert werden.

## Hinweise

### Validierung

Das W3C stellt online sowohl HTML-Validatoren <https://validator.w3.org> als auch CSS-Validatoren <http://jigsaw.w3.org/css-validator/> zur Verfügung. Beachten Sie, dass der Typ „date“ für Eingabefelder derzeit nicht in allen Browsern unterstützt wird. Eine entsprechende Warnung bei der Validierung dürfen Sie in diesem Fall ignorieren.

Zur Überprüfung der WAI-Tauglichkeit stehen Ihnen eine Vielzahl von Services im Internet zur Verfügung (z.B. AChecker, <http://achecker.ca/checker/index.php>). Nähere Infos dazu finden Sie in den Folien bzw. im TUWEL.

### Wahl der Elemente

Achten Sie darauf, semantisch sinnvolle HTML-Elemente zu benutzen. Überlegen Sie sich eine geeignete Repräsentation der Benutzer- und Produktinformationen. Tabellen zu Layout-Zwecken sind unter keinen Umständen erlaubt.

## Cascading Style Sheets (CSS)

Schreiben Sie das CSS vollständig selbst. Benutzen Sie also keine Frameworks wie Twitter Bootstrap, Pure, Foundation etc. Da Ihre Lösung nur in den aktuellsten Browsern funktionieren muss, können Sie für das Layout flexbox<sup>2</sup> oder float benutzen (oder beides).

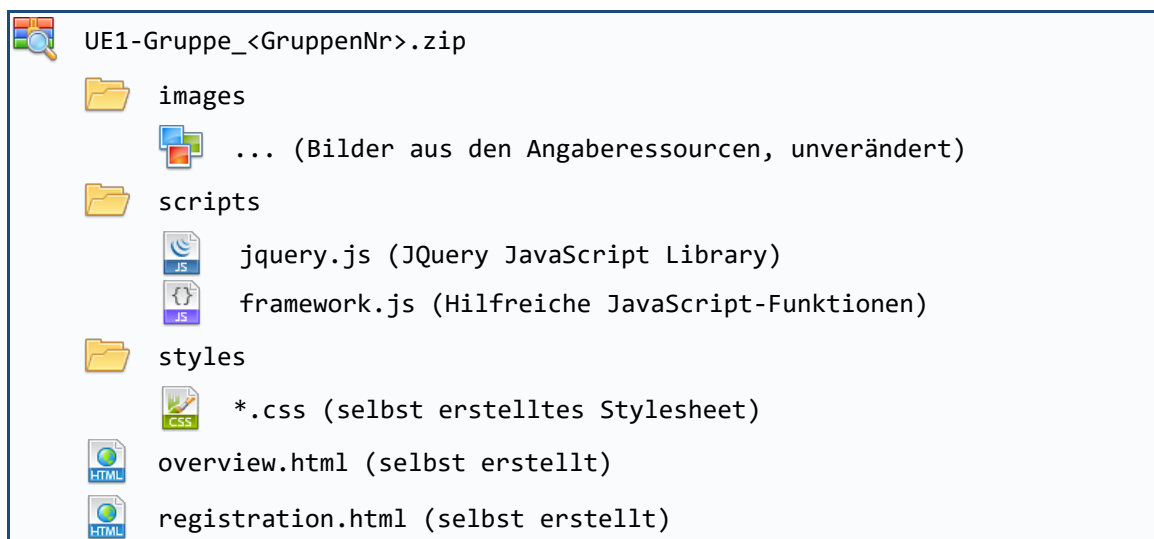
Benutzen Sie in Ihren Stylesheets die folgenden Werte als Breakpoints: 400px, 480px, 768px und 1024px.

## JavaScript

Zur Realisierung der Scripts darf nur jQuery<sup>3</sup> und die von uns bereitgestellte `framework.js` verwendet werden. Der Einsatz anderer JavaScript-Frameworks ist nicht gestattet. Eine Berücksichtigung der Barrierefreiheit der Scripts selbst wird für diese Übung nicht unbedingt gefordert, aber Sie sind eingeladen sich über diesen Aspekt selbstständig Gedanken zu machen und Lösungen zu realisieren.

## Abgabemodalität

Beachten Sie die allgemeinen Abgabemodalitäten des TUWEL-Kurses<sup>4</sup>. Zippen Sie Ihre Abgabe, sodass sie die folgende Struktur aufweist:



**Alle Dateien müssen UTF-8 codiert sein!**

**ACHTUNG:** Wird das Abgabeschema nicht eingehalten, so kann es zu Punkteabzügen kommen!

<sup>2</sup> <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

<sup>3</sup> <http://jquery.com/>

<sup>4</sup> <https://tuwel.tuwien.ac.at/course/view.php?id=7423>