

Web Technologien FS 2020 MEP: Bewertete Hausaufgabe

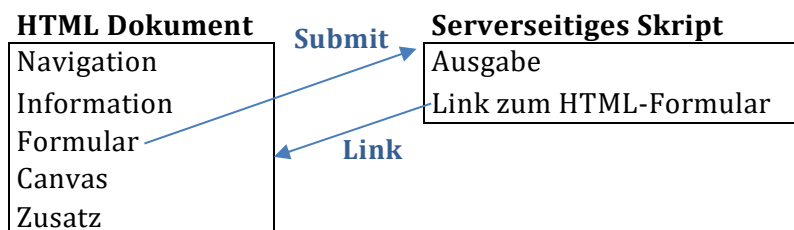
Die Modulendprüfung (MEP) findet im FS 2020 in Form einer bewerteten Hausaufgabe statt. **Dies ist eine Einzelaufgabe, d.h., Sie müssen diese Aufgabe selbstständig erarbeiten.** Diese Form des Leistungsnachweises ist eine von der HSLU gewünschte Alternative zur schriftlichen Prüfung¹, welche bisher für dieses Modul vorgesehen war. Sie haben einen Zeitraum von mehreren Wochen zur Verfügung um diese Hausaufgabe zu erarbeiten. **Diese Hausaufgabe ist einzureichen bis spätestens am:**

Freitag, 26. Juni 2020 um 23:59 Uhr.

1. Aufgabe

Erstellen Sie eine Webapplikation, welche aus einem HTML-Dokument und einem serverseitigen Skript besteht und die unterstehenden Anforderungen erfüllt. Die Webapplikation muss ein eigenes, einheitliches Thema haben. Der eigentliche Inhalt dieser Webapplikation ist Ihnen überlassen, muss aber zu dem gewählten Thema passen.

Die Struktur der Webapplikation (Darstellung im Webclient) sieht schematisch wie folgt aus:



2. Die Webapplikation muss folgende Anforderungen erfüllen:

- Die Applikation muss auf dem XAMPP-Webserver lauffähig sein.
- Es dürfen nur die im Kurs besprochenen Techniken (HTML, CSS, JavaScript, W3.CSS, Vue.JS, PHP, SQL, XML, JSON, AJAX) verwendet werden. Die Verwendung anderer Sprachen, zusätzlicher Funktionsbibliotheken oder Frameworks **ist nicht gestattet** und kann zum Abzug von Punkten führen.
- Sie wählen ein **eigenes** Thema für die Webapplikation. Sie dürfen hier kreativ sein. **ABER:** Das Thema darf keines der Themen sein, welche wir bereits in den Vorlesungsbeispielen oder Übungen behandelt haben (also kein Idealgewicht, Zufallszahl, Pizzashop, etc.).
- Das **HTML-Dokument** hat die folgenden vier Abschnitte:
 - Eine Navigationsleiste. Die Navigationsleiste soll ein direktes Anspringen der einzelnen Abschnitte (Information, Formular, Canvas, Zusatztechnik) innerhalb des HTML-Dokuments erlauben.

¹ Die HSLU sieht weitere Alternativen zur schriftlichen Prüfung vor, dies sind die mündliche Prüfung sowie eine elektronische Prüfung mit wenigen Fragen auf hohem Lernniveau. Diese beiden Formen des Leistungsnachweises eignen sich nicht für das Modul Web Technologien. Einerseits aufgrund der hohen Zahl an Teilnehmern des Moduls, andererseits aufgrund des Modulinhalts (breite Themenauswahl, wenig Tiefe bei den einzelnen Themen).

- Einen Informationsabschnitt, d.h., eine Anleitung, wie der Benutzer die Applikation zu bedienen hat. Dieser muss mindestens ein Paragraphenelement und ein Bild enthalten. Das Bild muss sich relativ zum HTML-Dokument in einem Unterverzeichnis mit dem Namen `img` befinden.
- Ein Formular mit mindestens drei Inputelementen unterschiedlichen Typs (type-Attribut), den entsprechenden Labels und einer Submit-Aktion. Die Submit-Aktion des Formulars muss das serverseitige Skript aufrufen und mindestens drei HTTP-Request-Parameter an dieses übermitteln.
- Einen Abschnitt mit einem selbstprogrammierten Canvas. Auf diesem Canvas müssen mindestens zehn Operationen ausgeführt werden (ausprogrammiert oder in einer Schleife). Als Operationen gelten beispielsweise `lineTo`, `arc`, `strokeText`, `fillRect`, `fillStyle`, `strokeStyle`, `beginPath`, usw.
- Verwenden Sie in Ihrer Webapplikation ausschliesslich relative URLs.
- Verwenden Sie jeweils passende HTML-Container (`<header>`, `<nav>`, `<section>`, etc.)
- Verwenden Sie mindestens vier CSS-Regeln, welche bei der Darstellung zur Anwendung kommen. Eine der Regeln muss einen Klassen-Selektor verwenden.
- Implementieren Sie **jeweils mindestens eine** Formularvalidierung:
 - deklarativ mit HTML
 - programmatisch mit JavaScript. Die programmatische Formularvalidierung muss den Inhalt eines oder mehrerer Inputelemente prüfen und darf keine deklarative Alternative haben. Beispielsweise hat die Prüfung auf ein leeres Eingabefeld eine deklarative Alternative und erfüllt diese Bedingung nicht.
- Das HTML-Dokument muss für mindestens zwei Abschnitte (z.B. Navigation und Information) über ein responsives Layout verfügen und für die zwei Geräteklassen Mobile und Desktop unterschiedliche Darstellungen verwenden. Dies muss unter der Verwendung von W3.CSS erfolgen. Ein Beispiel für den Informationsabschnitt wäre die Darstellung von Bild und Text nebeneinander in einer Desktopauflösung und ansonsten gestapelt. Analog für Labels und Inputs im Formularabschnitt.
- Das **serverseitige Skript** soll folgende Funktionalität implementieren:
 - Existenzprüfung von mindestens drei HTTP-Request-Parametern.
 - Das serverseitige Skript soll eine HTTP-Response mit Inhalt von Typ HTML erzeugen, welche mindestens drei HTTP-Request-Parameter verarbeitet.
 - Die HTTP-Response muss einen Link zurück zum HTML-Dokument haben.
 - Das serverseitige Skript muss ein Cookie setzen, welcher zum Thema passt.
- ***Eine*** Zusatztechnik (keine Zusatzpunkte beim Implementieren von mehr als einer Technik). Implementieren Sie genau eine Technik aus der folgenden Liste von Techniken:
 - a) Als Reaktion auf eine Benutzeraktion soll mittels AJAX ein JSON- oder XML-Dokument abgeholt werden und mindestens eine Information aus diesem Dokument auf dem HTML-Dokument eingeblendet werden.
 - b) Serverseitige Abfrage von Daten aus einer MySQL-Datenbank und Ausgabe der Daten im serverseitigen Skript. Das in phpMyAdmin ausführbare Datenbankschema muss in `schema.sql` mitgeliefert werden und Tabelleneinträge beinhalten. Verwendung von Prepared-Statements zwingend.
 - c) Serverseitiges Einfügen von Daten in eine MySQL-Datenbank. Das in phpMyAdmin ausführbare Datenbankschema muss in `schema.sql` mitgeliefert werden. Verwendung von Prepared-Statements zwingend.

- d) Abfragen der aktuellen Geräteposition als Reaktion auf Benutzeraktion und einfügen der Position in ein Inputfeld des Formulars, welches zum serverseitigen Skript übermittelt wird. Im serverseitigen Skript soll der Wert als funktionierender Link zu GoogleMaps (oder einem alternativen Kartendienst) dargestellt werden.
 - e) Implementation einer clientseitigen, reaktiven Berechnung mittels Vue.js unter korrekter Verwendung von mindestens vier unterschiedlichen Vue.js Direktiven bzw. Templates. Beispiel von vier unterschiedlichen Direktiven: v-model, v-bind, v-if, {{ ... }}.
- **Abgabeformat:** Das Abgabeformat muss exakt eingehalten werden. Die Abgabe erfolgt als ZIP-Datei mit Namen WEBT_FS20_<NACHNAME>_<VORNAME>.zip.
Beispiel: WEBT_FS20_BAETTIG_MARTIN.zip.

Die Zip-Datei muss **exakt** folgende Verzeichnisstruktur aufweisen:

- <NACHNAME>_<VORNAME> (Beispiel: BAETTIG_MARTIN, Verzeichnis in Grossbuchstaben)
 - HTML-Dokument (Name beliebig)
 - PHP-Skript (Name beliebig)
 - img (alle Bilder in dieses Verzeichnis)
 - Bild 1 (Name beliebig)
 - Bild 2 (Name beliebig)
 - ...
 - schema.sql (nur falls SQL-Datenbank benötigt).

3. Abgabe

Die Abgabe erfolgt im ILIAS im entsprechenden Briefkastenordner **für ihr Modul:**

- I.BA_WEBT_MM.F2001/03 Abgabe Hausaufgabe
- I.BA_WEBT_MM.F2002/03 Abgabe Hausaufgabe
- I.BA_WEBT_MM.F2003/03 Abgabe Hausaufgabe
- I.BA_WEBT_MM.F2004/03 Abgabe Hausaufgabe

Folgende Punkte sind bei der Abgabe zu beachten:

- Frühere Abgaben sind möglich.
- Mit der Abgabe geben Sie Ihre Hausaufgabe zur Bewertung frei.
- **Abzug bei verspäteter Abgabe: pro Tag eine halbe Note**

Im Notfall, d.h., wenn ILIAS zum Abgabezeitpunkt nicht erreichbar ist, können Sie Ihre Abgabe per Email an folgende Adresse schicken: martin.baettig@hslu.ch.

4. Plagiate

Dies ist eine Einzelarbeit. Sie müssen diese selbstständig erarbeiten. Wir haben relativ viel Erfahrung mit Plagiaten und sehen z.T. Details, welche Sie beispielsweise Lösungen von Mitstudenten oder Code aus dem WWW übernehmen und anpassen, nicht sehen werden. **Bitte arbeiten Sie daher selbstständig. Der Ärger ist es nicht wert, für beide Seiten!**

Viel Erfolg wünschen Ihre Dozenten
Martin Bättig (MV), Raphael Ritter und Martin Zimmermann

5. Bewertungsraster

Nachfolgend finden Sie das Bewertungsraster. **Für eine genügende Note müssen Sie mindestens 22 Punkte erreichen.**

Bewertungskriterium	Maximale Punkte
Navigationsleiste <ul style="list-style-type: none"> - Vier interne Links vorhanden, welche zu den einzelnen Abschnitten des HTML-Dokuments führen. 	2
Informationsabschnitt <ul style="list-style-type: none"> - Mindestens ein Paragraph-Element vorhanden. - Mindestens ein Bild-Element vorhanden. 	2
Formular <ul style="list-style-type: none"> - Mindestens drei Inputelemente unterschiedlichen Typs (type-Attribute) mit entsprechenden Labels vorhanden. - Submitaktion mit korrekter Zielaktion vorhanden. - Drei Request-Parameter übermittelt. 	3
HTML-Container <ul style="list-style-type: none"> - Verwendung des passenden Containertyps (<header>, <nav>, <section>, etc.). 	2
CSS <ul style="list-style-type: none"> - Definition von vier CSS-Regeln. - Die definierten Regeln kommen je mindestens einmal zur Anwendung. - Mindestens ein CSS-Selektor selektiert auf eine Klasse, welche existiert. 	2
Responsives Layout mittels W3.CSS <ul style="list-style-type: none"> - Mindestens zwei Abschnitte verfügen über ein responsives Layout. (Responsive: Breakpoints sind definiert, z.B., unterschiedliche Grids für Mobile/Desktop. Das Aussehen selbst wird nicht bewertet.) 	2
Formularvalidierung <ul style="list-style-type: none"> - Eine deklarative Validierung ist vorhanden. - Eine fachliche Validierung mittels JavaScript ist vorhanden, welche nicht durch eine deklarative Validierung ersetzt werden kann. 	3
Selbstprogrammiertes Canvas-Element <ul style="list-style-type: none"> - Inhalt wird korrekt dargestellt. - Es wurden mindestens zehn Operationen auf dem Canvas ausgeführt (ausprogrammiert oder als Schleife). 	2
Serverseitiges Skript <ul style="list-style-type: none"> - Drei Parameter überprüft auf Existenz. - Drei Parameter sind Teil einer Operation (Stringverkettung, Berechnung, etc.). - Anzeige des Resultats basiert auf den übermittelten Werten. - Link zurück zum Formular. 	4
Cookie <ul style="list-style-type: none"> - Setzen eines Cookies durch das serverseitige Skript. 	1
Zusatztechnik <p>a) Webservices: Eine AJAX-Anfrage (Reaktion auf Benutzereingabe), Verarbeitung der Antwort (XML- oder JSON-Format), Einfügen im HTML-Dokument.</p>	3

<p>b) Ein eigenständiges SQL-Schema mit Tabelleneinträgen, <i>schema.sql</i> ausführbar in phpMyAdmin, Daten abgefragt und angezeigt, Prepared-Statements verwendet.</p> <p>c) Ein eigenständiges SQL-Schema mit <i>schema.sql</i> ausführbar in phpMyAdmin, Daten eingefügt mittels serverseitigem Skript, Prepared-Statements verwendet.</p> <p>d) Geolocation: Abfrage der Geräteposition (Reaktion auf Benutzereingabe), Übermittlung an den Webserver via Formular, Anzeigen des Karten-Links zur Geräteposition.</p> <p>e) Vue.js: Mindestens vier unterschiedliche Vue.js Direktiven korrekt angewendet, reaktive Berechnung.</p>	
<p>Thema</p> <ul style="list-style-type: none"> - Konsistente Verwendung des Themas über alle Teile (Navigation, Information, Formular, Canvas, Serverseitiges Skript, Zusatztechnik) der Webapplikation hinweg. - Kein Thema aus den Vorlesungsbeispielen und dem Übungsmaterial wiederverwendet. 	3
<p>Abgabeformat</p> <ul style="list-style-type: none"> - Durchgängig relative URLs verwendet. - Als ZIP-Datei mit korrektem Namen. - Korrekte Verzeichnisstruktur. - Auf XAMPP lauffähig. 	4
Total	33

6. Anhang

Beispiel eines in phpMyAdmin ausführbaren *schema.sql* (copy und paste in das SQL-Tab und click auf OK):

```
create database tasks;
use tasks;
```

```
create table tasks(user varchar(200), task varchar(1000));
insert into tasks (user, task) values ('jack', 'Hausaufgaben fuer Dienstag erledigen');
insert into tasks (user, task) values ('kim', 'Projektarbeit fertigstellen');
insert into tasks (user, task) values ('luke', 'Auto waschen');
insert into tasks (user, task) values ('joe', 'Geburtstagsparty organisieren');
insert into tasks (user, task) values ('kim', 'Unittests schreiben');
insert into tasks (user, task) values ('jack', 'Hausaufgaben fuer Donnerstag erledigen');
```